RESEARCH TRIANGLE INSTITUTE

NASA CR-112195

DEVELOPMENT OF SIMULATION TECHNIQUES SUITABLE FOR THE ANALYSIS OF AIR TRAFFIC CONTROL SITUATIONS AND INSTRUMENTATION

FOR U. S. GOVERNMENT AGENCIES AND THEIR CONTRACTORS ONLY

Prepared Under Contract No. NAS1-10847

by

Research Triangle Institute
Research Triangle Park
North Carolina 27709

RTI No. 43U-718

for

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23365

December 1972

ACKNOWLEDGEMENT

This report was prepared by the Research Triangle Institute, Research Triangle Park, North Carolina, under Contract NAS1-10847. The work is being administered by the Flight Instrumentation Division, Langley Research Center. Mr. Leonard Credeur is the Langley Technical Representative for the contract.

For the studies described in this report, work has been closely coordinated with personnel at Langley Research Center. Messrs. T. Walsh, L. Credeur, D. Cannon, and W. Capron have contributed heavily to the modeling and computer simulation work described herein.

RTI staff members participating in the study were as follows:

C. L. Britt, Project Leader
E. G. Baxa, Systems Engineer
J. Modi, Operations Analyst
R. C. Haws, Systems Analyst
C. M. Davis, Systems Analyst
J. L. Gatz, Research Assistant.

Preceding page blank

TABLE OF CONTENTS

PRECEDING PAGE BLANK NOT FILMED

TABLE OF CONTENTS, Continued

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS, Continued

LIST OF ILLUSTRATIONS, Continued

Preceding page blank

LIST OF TABLES

PRECEDING PAGE BLANK NOT FILMED

SUMMARY

This report describes a terminal area simulation which permits analysis and synthesis of current and advanced air traffic management system configurations including ground and airborne instrumentation and new and modified aircraft characteristics. Ground elements in the simulation include navigation aids, surveillance radars, communication links, air-route structuring, ATC procedures, airport geometries and runway handling constraints. Airborne elements include traffic samples with individual aircraft performance and operating characteristics and aircraft navigation equipment. The simulation also contains algorithms for conflict detection, conflict resolution, sequencing and pilot-controller data links.

The simulation model will be used to determine the sensitivities of terminal area traffic flow, safety and congestion to aircraft performance characteristics, avionics systems, and other ATC elements. These results will assist in establishing future requirements for (1) basic sensor/instrumentation accuracies for airborne and ground systems; (2) permissible system delays in airborne and ground information handling systems; (3) required data rates for air-ground information exchange; (4) requirements for signal and data processing for both airborne and ground computers; and (5) aircraft performance and operating characteristics.

The initial model has been programmed for the CDC 6600 computer and run in both real-time and fast-time on the computer complex at Langley Research Center. The initial test runs indicate that the model and associated program are within the capabilities of the computer facilities available. The test runs also indicate good correlation with actual radar traffic data obtained by the FAA at the Atlanta airport.

II. INTRODUCTION

The development of air traffic control systems is a continuous evolutionary process, with new developments being constantly spurred by rapidly changing aviation technology and the demands of the public at large. However, changes in any part of the air traffic control system are bound to have complex effects in several aspects of system operation. Computer simulation has proved to be an effective means of evaluating the impact of proposed improvements on overall system performance.

This report describes the development of, and initial experiments with a computer simulation model of a terminal area. The primary objective under this contract has been to develop a preliminary mathematical model of the Atlanta terminal area, to program the model for the Langley Research Center digital computer complex, and to demonstrate the operation of the preliminary model on the LRC computers. The computer model is intended to be flexible and extendable to other terminal areas and to terminal area configurations using advanced concepts.

The basic concepts for the simulation model were defined and described in detail under Contract NAS1-7537. The final report developed under this contract, "Definition of a Terminal Area Air Traffic Model for Studies of Advanced Instrumentation and Control Techniques" [Ref. 1], contains discussions of the model parameters considered and sources of data that provide inputs to the model.

The initial work described in Ref. 1 indicated that the model and associated program were within the capabilities of the computer facilities available; however, for efficient use of these facilities it was highly desirable to reduce the computer storage requirements. This reduction in storage was advisable in order to incorporate more detail and flexibility in certain areas of the program. For this reason, a large part of the effort in this study was devoted to providing more efficient use of the computer memory by densely packing the input data into the 60-bit words of the CDC-6600 computer.

A significant number of modifications have been made to the initial model defined in Ref. 1. These new features are intended to increase the realism and flexibility of the basic model, its adaptability to different and advanced terminal areas, and its versatility in the

investigation of different types of problems. The major new features introduced into the model include:

Conflict Detection and Resolution

A fully dynamic technique of conflict detection and resolution has been incorporated whereby certain aircraft are rechecked for clearances depending on the action of an aircraft ahead.

Route Structure

More flexible route structure design is permitted providing choices of controller conflict resolution options as designated by input data.

Sequencing

Previously, all aircraft entered the sequencing area via a single feeder route in each sector (North and South). Final sequencing logic has now been modified to allow for multiple merging feeder paths into each sector sequencing area. Final sequencing is accomplished for each aircraft on its respective feeder path at designated points such that as the aircraft merges into the dump area no loss of separation is encountered. Additionally, the new logic permits more realistic interception of the localizer course.

Control Positions

Provision has been made to provide any number of simulated control positions.

Altitude Conflict Checks

Additional logic has been provided to check for conflicts between departing aircraft and arriving aircraft at places where their future paths may cross.

Cockpit and Runway Display

Coding has been provided such that a general-purpose cockpit simulator can be used in conjunction with the air traffic model. A graphic runway display (on an oscilloscope) indicates in real time the runway in perspective as it would appear to a pilot in a cockpit simulator.

Pop-Ups

A capability has been incorporated to simulate pop-up traffic, e.g., a sudden appearance of unknown traffic anywhere on the simulated radar screen. Pop-ups are simulated by entering the simulated pop-up

aircraft's position and velocity data on the computer console at the time of desired entry.

Navigation and Surveillance Errors

More realistic representation of these errors has been incorporated.

Speed Control

Algorithms have been developed to provide a speed control feature for resolving conflicts. Input data define the type and range of speed control to be used.

Missed Approaches

Algorithms have been developed to reintroduce missed approaches and wave-offs into the traffic pattern for another attempt at landing. These algorithms, however, have not yet been incorporated in the computer program.

The following sections of the report discuss the above model features in more detail and provide documentation of the application of the model to the Atlanta terminal area.

III. GENERAL DESCRIPTION OF THE SIMULATION MODEL

A. Overall Organization

The overall simulation model consists of three distinct parts: (1) a traffic generation computer program, (2) the terminal area simulation model program, and (3) the data analysis program. In addition, provision is currently being made for the incorporation of a general-purpose aircraft cockpit simulator and associated computer program. The overall model organization is indicated in Fig. 1.

For operation in the real-time mode, the traffic sample is generated in a batch mode prior to the real-time runs, and data analysis is accomplished by storing simulation program outputs on tape for later non-real-time analysis.

B. Traffic Generation

A Monte Carlo program is used for generating data for the traffic sample. Arrivals are introduced into the problem at a 40 n mi radius from the airport and departures are introduced at the runway apron. From a probability distribution of time-between-operations, an offer time is selected for the next aircraft to be provided in the traffic sample. For the aircraft to be entered, a type is selected from a histogram of aircraft types. Next, the program decides whether the aircraft is an arrival or departure and what route the aircraft will be introduced on. After this information has been calculated, an altitude, speed, and bearing from the terminal area are calculated using normally distributed deviations about the nominal route assignments. After calculation of the x and y coordinates, a heading is calculated such that the aircraft will be headed toward the assigned (from the known route) inner fix.

Using this technique, a traffic sample is generated which has statistics corresponding to those statistics that provide the inputs to the program. The traffic sample may be as lengthy as required. For real-time simulations, a one-hour traffic sample is usually generated.

Major inputs to the traffic sample generation program include the desired number of operations per hour, the number of aircraft types, the desired sample length (min), probabilities of aircraft type and route usage,

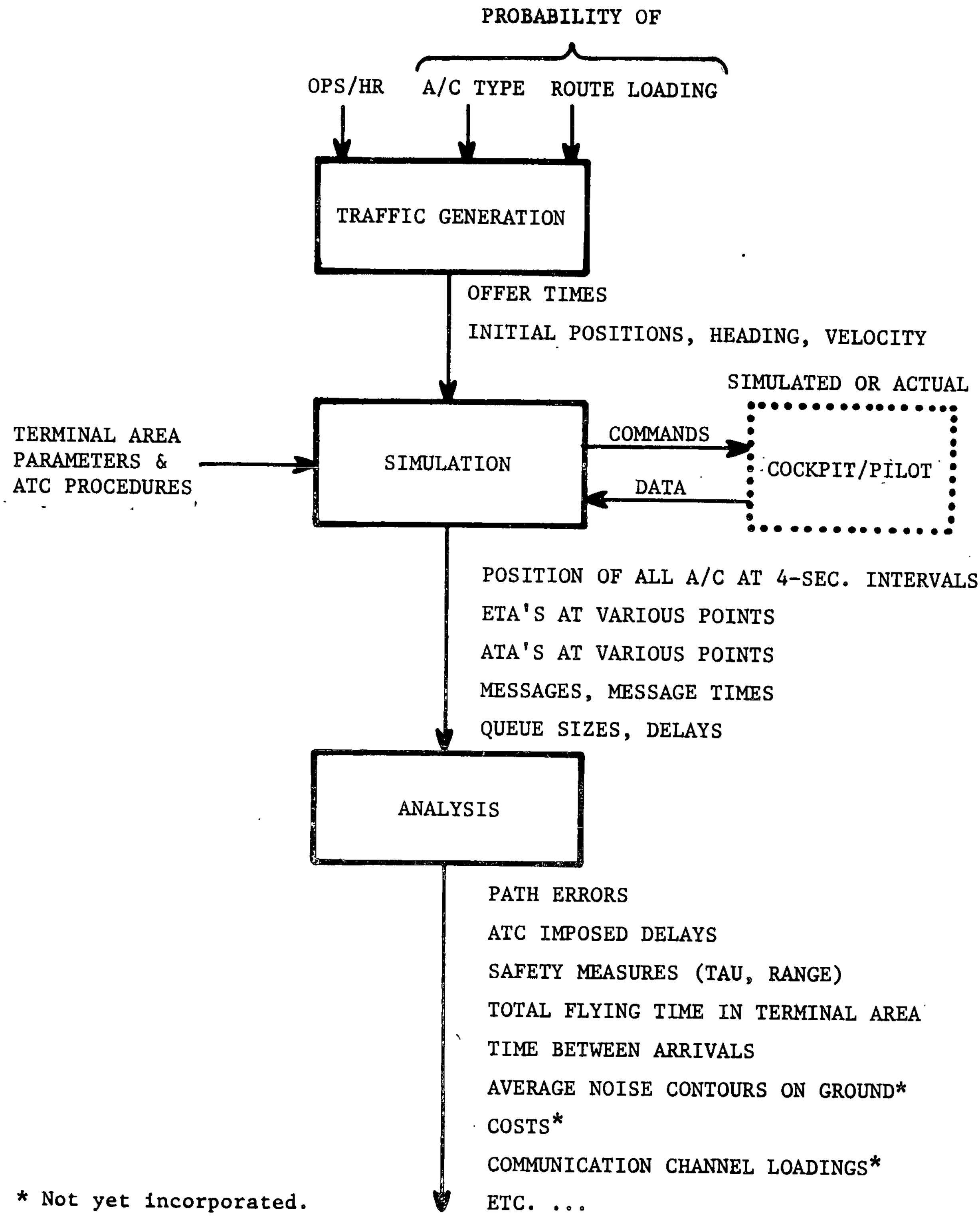


Fig. 1. Overall organization of the terminal area simulation model.

route entry coordinates, and standard deviations of initial errors in altitude, speed, and position.

Output of the traffic generation program is a list of aircraft with the initial conditions (i.e., offer time and position) for each aircraft, as well as type and class identification.

C. Terminal Area Simulation

A functional flow chart indicating the overall organization of the simulation model is shown in Fig. 2. Inputs to the simulation include arrays defining aircraft performance, the route structure, the traffic sample, intended flight plans, separation standards, navigation errors, communication message times, and control procedures and options.

From the offer times in the traffic sample, an active list of N aircraft is generated for further processing. Each aircraft entering the problem has associated with it a nominal route or intended flight path (IFP) which represents the path the pilot would fly if no conflicts existed with other traffic. Various legs along the flight paths are defined by flight modes and objectives. The modes are generally defined by the navigation technique in use. For example, algorithms have been developed for flying a VOR radial, a DME radius, a vector, an ADF beacon, an ILS system, a holding pattern, and a standard turn. Objectives are also, in most cases, defined by the navaids. Examples of objectives are a VOR radial intersection, a particular fix or geographic point, a DME radius, a new altitude, etc.

For each of the N aircraft on the active list, the current flight mode and next objective are determined from the intended flight path file. Each aircraft is then incremented one step along its path using a set of aircraft dynamic equations which contain the dynamic constraints on the path. As each of the active aircraft progresses along its flight path, a test is made to see if the aircraft's current objective has been reached. If so, the next flight mode is selected and the process continues.

At certain regions along the nominal flight paths, a test is made for potential conflicts at other regions along the intended paths. If a conflict is detected, it is resolved by appropriate action such as a speed, altitude or path change, or by a holding maneuver. The conflict resolution or other control action is associated with a control message generated

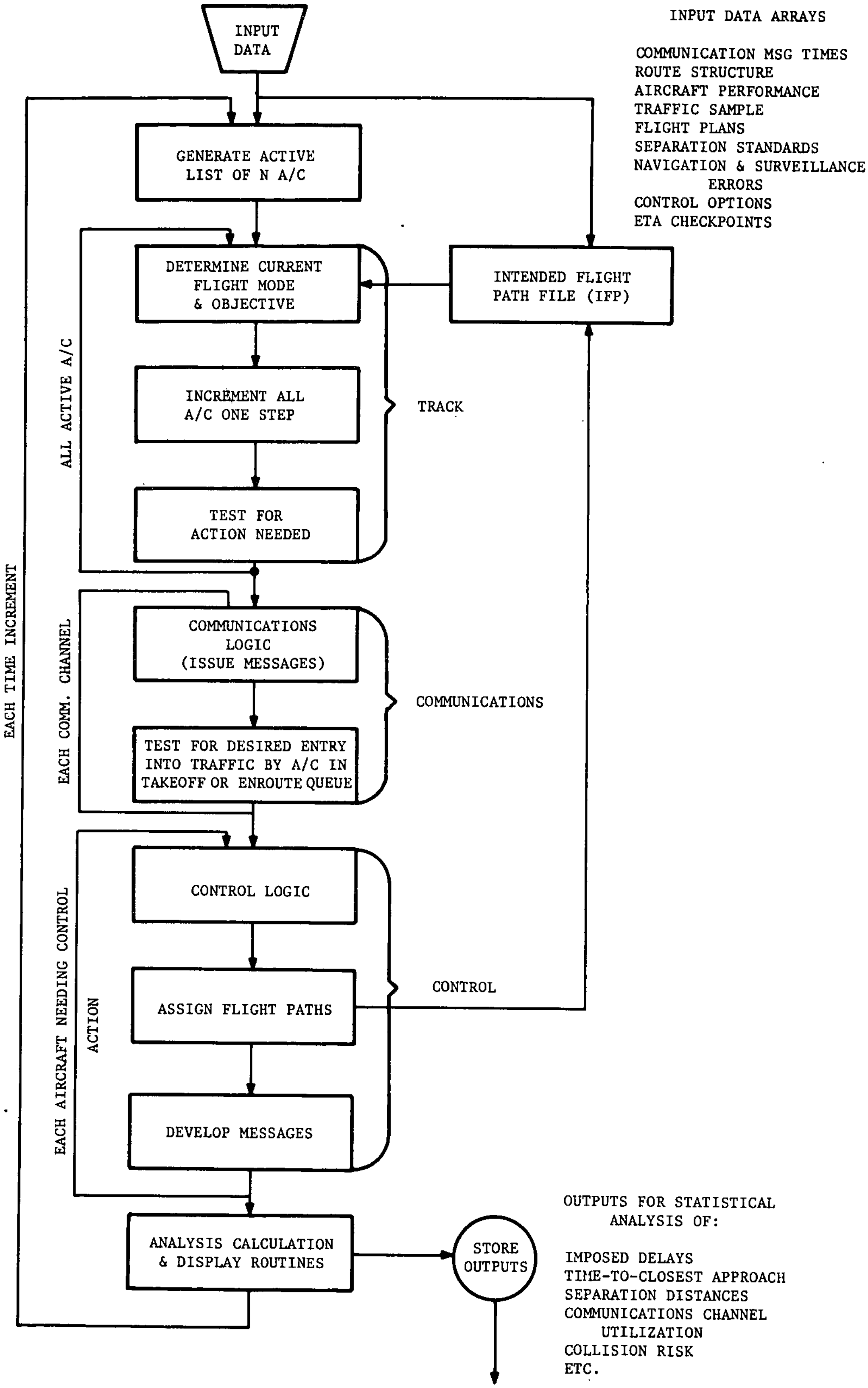


Fig. 2. Functional flow chart--overall organization of simulation program for the initial model.

on the applicable communications channel to initiate a change in the flight mode (e.g., a new vector or a speed change) at the appropriate time, or to give clearance on a desired path. The messages have associated with them an earliest and latest possible transmission time and are issued within this time frame as communications channel loading permits. For the initial model, ten controller positions and associated communications channels are modeled. Provision is also made to handle "pop-ups," or aircraft that are inserted (by the computer console operator) at randomly selected positions and times. These pop-ups are integrated into the normal flow patterns in a conflict-free manner.

After all aircraft needing control action have been dealt with, the information on the present position of all aircraft is displayed on a computer console display and x-y plotters as required. The time histories of the takeoff queues, arrival queues, and communications channel queues are recorded on strip chart recorders. The basic data generated and stored on tape for further analysis include the position and velocity of each aircraft at selected time increments throughout the period of interest, the messages sent over each communication channel, and the estimated and actual times of arrival of the aircraft over certain specified points along the routes assigned to the aircraft.

D. Data Analysis

Techniques have been developed that permit the generation of statistics such as "the average percentage of flying time in the terminal area that a randomly selected aircraft is within a range R of another aircraft." A statistic such as this when plotted versus various values of range provides a good measure of the efficiency of the separation procedures used. Under reasonable assumptions, the percent of flying time statistic also provides an unbiased estimate of the probability that a randomly selected aircraft will be within a range R of another aircraft at any particular instant of time under conditions existing during the data period.

While the parameter "range to closest aircraft" is discussed in the above paragraph as an example, other parameters can be investigated in a similar way. For example, a parameter indicating the safety level of the system during a period of operation would be "time to closest approach," or relative range divided by closing velocity. In this case, a distribution

plot is formed of the average percentage of flying time in the terminal area that a randomly selected aircraft is within a time to closest approach of τ seconds of another aircraft.

In addition to the development of statistics as discussed in the examples above, the conventional output measures can also be developed. These measures include statistics such as the distribution of control-related delays, deviations from original ETA's, average mileage flown, aircraft movement rates, etc.

E. Detailed Procedures

Additional discussions of model philosophy and operation are given in Refs. 1 and 2. Specific computer algorithms are given in general flow chart form in Appendix A and in detailed (computer coding) flow chart form in Appendix H. Appendices B and C contain the input data specifying the parameters of the Atlanta simulation. The major subroutines and model variables are defined in Appendices D and E, respectively. Appendix F provides a description of the packed computer words and Appendix G specifies the computer storage requirements for the model as it exists to date.

The detailed procedures for the use of the model are best illustrated by an example of the application of the model to the simulation of the Atlanta terminal area. The application to the current Atlanta terminal procedures is presented in the following section of this report.

IV. APPLICATION OF THE MODEL TO THE ATLANTA TERMINAL AREA

A. General Procedure

For the application of the simulation model to a specific terminal area, the general procedure consists of several steps which may be generally outlined as follows.

1. Data sheets defining the performance characteristics of each aircraft type to be simulated are filled out.

2. On a large sheet of graph paper (scale approximately 5 n mi/in), the position of the terminal area navigation aids and the airport runways are plotted. These aids include the VOR/DME locations, ILS locations, non-directional beacon coordinates, and important navigational fixes. The data sheet associated with the navigation aids is filled out.

3. Using the navigation aids and simulated controller vectors, the nominal arrival and departure routes are defined. The altitude and speed profiles along the routes are defined. Associated data sheets are filled out.

4. Again using the navigational aids and simulated controller vectors, alternate routes (if desired) are developed that will be used to resolve conflicts where applicable. The nominal and alternate routes are generally plotted in straight-line (or curved) segments on the large sheet of graph paper, with arrival and departure routes plotted separately.

5. Points along the nominal and alternate routes at which conflict checks will be made are specified. These points are designated as Conflict Checkpoints. In addition, regions of airspace where routes (e.g., arrivals and departures) may intersect are noted on the plots.

6. Points along the nominal and alternate routes at which a specified control action will take place are defined. These points are designated as Control Action Points.

7. For each Control Action Point and associated Conflict Checkpoint, the options that will be used to resolve a conflict are selected from the list of available options. The associated input data sheets are filled out.

8. Miscellaneous terminal area procedures such as separation standards, controller responsibility areas, aircraft dynamic parameters, glide slope angles, and other miscellaneous data are specified and listed

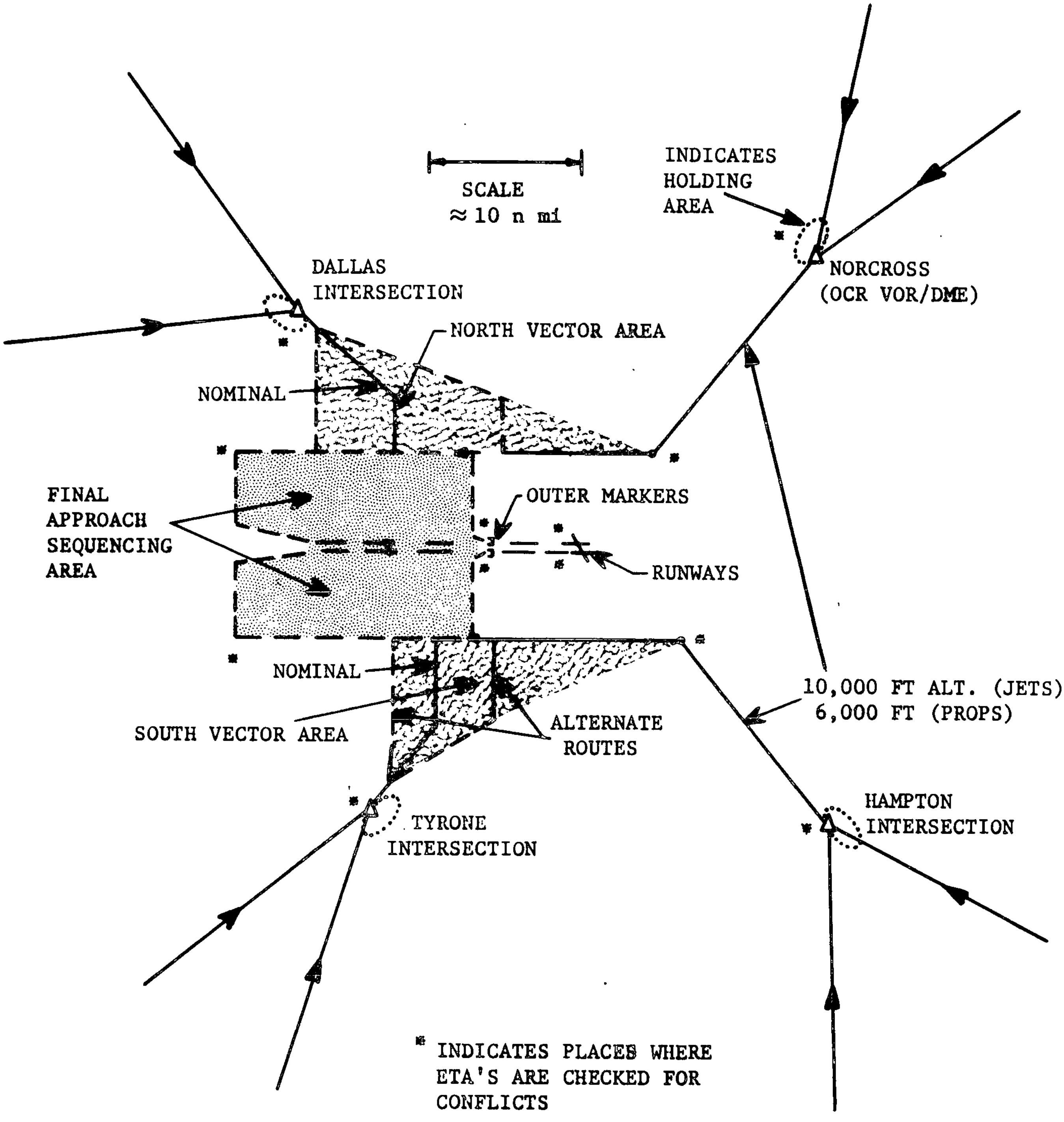


Fig. 3. Sketch of arrival routes defined for initial simulations. In the final approach sequencing area, the longest and shortest paths are shown as dashed lines. Asterisks indicate where ETA checkpoints are specified.

on the appropriate input data sheets.

9. The traffic sample generation program inputs are defined by developing probabilities associated with the route usage, the aircraft type utilization, and the number of operations desired per hour.

The above general outline indicates in simplified form the procedures followed. The specific procedures are best illustrated by use of the example of an application to the Atlanta terminal area as presented in the following sections.

B. Aircraft Performance Data

The data sheets specifying the aircraft performance parameters required for the simulation are given in Appendix B, page 124. These parameters include, for each aircraft type, data on climb/descent rates, acceleration and deceleration characteristics, speed characteristics, and runway occupancy information.

C. Atlanta Navigational Aids

The navigational aids available in the Atlanta area are listed on the data sheet given in Appendix B, page 126. For each type of navigational aid, the x-y coordinates, the rms instrument error, and a "make-good radius" which defines a circle within which an aircraft will be considered to have reached the navigational site, are listed. The coordinates are referenced to the location of the Atlanta surveillance radar, which is located between the two parallel east-west runways at Atlanta.

D. Atlanta Route Structure

The term "route structure" as used here refers not only to the layout of the flight paths in terms of their segments, orientations, altitudes, etc., but also to the system of conflict check points and controller action points by which the air traffic control function is simulated. The basic schema of the route structure is similar to that described in Ref. 1, in that the same "four-poster" pattern is used for arrivals, and that departures are routes in the same general directions. However, there are important differences in the detailed layouts of the paths, particularly vector paths, and in the actual mechanics of simulating the control function.

The physical layout of the flight paths used in the current Atlanta terminal area simulation is shown in Fig. 3. It is designed to approximate

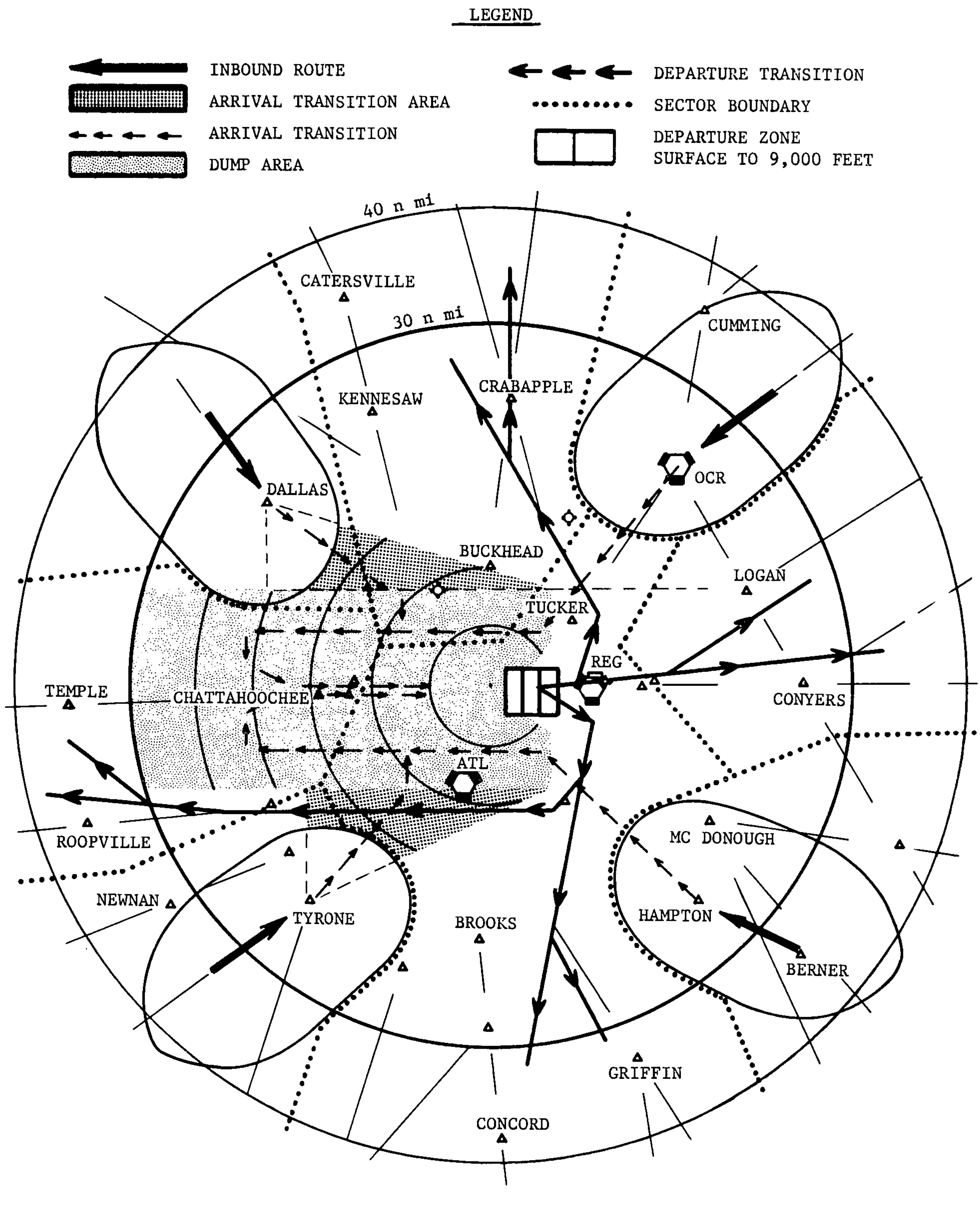


Fig. 4. Chart showing general arrival and departure routes for runways 9L and 9R at Atlanta. These data are given in the Atlanta Tower Orders for each runway.

patterns and procedures represented in Fig. 4, which is part of the Letter of Agreement between Atlanta Towers and Atlanta Center that went into effect November 11, 1971. Referring, first, to Fig. 4, arrivals enter the terminal area at the four initial approach fixes (IAFs) DALLAS, NORCROSS, HAMPTON, and TYRONE. There is one initial approach course (shown by a short, heavy arrow) for each IAF, which corresponds roughly to the published standard terminal arrival routes (STARs) for Atlanta. Jet aircraft are assigned altitudes 10,000 or 11,000 ft, and propeller aircraft are assigned 6,000 ft in the terminal area. At all four IAFs, the STARs require aircraft to be at their assigned altitudes about 10 n mi before they reach the IAF, and (for jets) to begin speed reduction to 250 kts when they reach assigned altitude.

At some point before they reach the IAFs, aircraft are handed over by the Center to Atlanta Approach Control. Beyond the IAFs, two basic patterns are followed. Aircraft over NORCROSS and HAMPTON are given one vector to leave the IAFs (220° and 320°, respectively), and speed is not changed unless necessary; when necessary, a typical speed is 200 kts. Aircraft over DALLAS and TYRONE may get two or three vectors in addition to the above speed control in an effort to merge them with aircraft over NORCROSS and HAMPTON respectively. The vectoring area is shown cross-hatched in Fig. 4.

Figure 4 also shows an area known as the dump area. Its northern and southern boundaries are defined by the Fulton County VOR 90°-270° radials and the Atlanta VOR 90°-270° radials, respectively. The dump area contains the downwind legs, where all aircraft are descended and merged in preparation for their sequencing for final approach. Typically, aircraft coming from DALLAS and TYRONE are turned heading 180° and 360° respectively, and slowed to 170 kts as they enter the dump area; similarly, aircraft coming from NORCROSS and HAMPTON are turned heading 270° and slowed to 170 kts as they enter the dump area at its eastern edge. Soon after entering the dump area, the aircraft are under the jurisdiction of the final controller, who turns aircraft onto the base leg and then onto the final approach course in proper sequence. Once established on the final approach course they are asked to monitor the tower frequency on which the local controller clears them to land.

Departure procedures are somewhat simpler because much greater freedom for vectoring is available. Figure 4 shows what are essentially nominal departure paths in four major compass directions, but a large airspace is

available for departures, allowing great freedom in headings, altitudes and speeds. The main points to note are that: (1) westbound departures must turn right only, and (2) all but eastbound departures cross arrival routes. Ordinarily, jets are capable of climbing above the arrival routes and are so cleared, but prop aircraft might need to climb through some arrival altitudes. This is permitted if proper coordination can be effected in advance; if not, the climb is restricted to one of the interleaved departure altitudes until the aircraft is past the arrival transition area.

The route structure used in the simulation model corresponds to the above patterns and procedures. Aircraft approach the same four IAFs on two initial approach courses each, as shown in Fig. 3, with one of the courses at each fix corresponding to the published STAR procedures. Vectoring of arrivals coming over DALLAS and TYRONE is simulated by three paths of four segments each in each of the vectoring areas; these are the paths 7, 9, and 10 out of DALLAS and paths 17, 19, and 20 out of TYRONE. Altitude and speed control procedures are as described except that the simulator permits jets at only one altitude (10,000 ft) from the IAFs to the dump areas. Departure procedures are also as described above, except that departure vectoring is restricted to the nominal headings shown in Fig. 4.

Flight Schedule (ISKED) Input Data

Data input to provide the route structure definition is listed in Appendix B, pages 127 to 131. This particular data array is called the Flight Schedule (ISKED) array or intended flight path (IFP) array, and defines the flight modes and flight objectives for each segment of flight in terms of the navigation aids and controller vectors. Each segment may be a straight and level path, an accelerating straight climbing path, a curved climbing path, etc.

The design of the IFP array is intended to accommodate the requirements of the aircraft dynamics or TRACK algorithms (see Fig. 2). In the TRACK algorithms, flight paths are computed in different ways, depending on which "mode" the flight is in. Therefore, the first field in the IFP record consists of the flight mode. Flight modes are designed to require at most two parameters to describe flight; these are supplied as INFO1 and INFO2. The codes for the flight mode and the information to be inserted

in the two fields for various flight modes are indicated in Table 1.

Termination of a particular leg is indicated by information on the objective of the flight leg. Objectives also require at most two parameters, INFO3 and INFO4, to describe the termination of the leg; the codes for the leg objectives and the information to be inserted in the two fields are indicated in Table 2.

In addition to the Flight Modes and Flight Leg Objectives, information is also provided in the Flight Schedule array to specify the desired altitude and speed of the aircraft. The information is in the form of special codes and pointers as shown in Table 3.

The above concepts are illustrated by correlating the instructions given in the Atlanta flight schedule array (Appendix B, page 127) with a sketch of a portion of the routes given in Fig. 3. This correlation is shown in Fig. 5, where a segment of the route is re-sketched along with the corresponding items in the Atlanta flight schedule array.

Referring to Fig. 5, the first line in the flight schedule array specifies that the first leg will be flown using a VOR mode (Code 1) and that the VOR used will be OCR (Code 3 in the INFO1 field identifies the OCR VOR as specified on the Navigational Aid data sheet) with a 12° (INFO2) radial from this VOR. The next data item is an altitude pointer which indicates to the program where to find the altitude data for this leg. The value 31 (see Table 3) is a special code indicating that the desired altitude for this leg will be found in the altitude option array (Appendix B, page 144). Physically, this represents the fact that the controller rather than the pilot will choose the altitude.

Following the altitude pointer is a speed pointer which indicates the type of speed desired on this leg. The pointer points to the corresponding column of the speed performance array (IPERFS) which is given in Appendix B, page 124. The special code 0 indicates that on this leg the aircraft should continue at its initial descent speed (e.g., no speed change is indicated on this leg unless dictated by controller action).

The next three fields of data indicate the objective of this flight leg. The mode 3 code (see Table 2) indicates that the objective for the first leg is a specific distance from the initial leg point of 9 n mi (INFO3).

The remaining lines in the schedule array define the other legs shown in Fig. 5. Physically, the meaning of each line shown in Fig. 5

Table 1. Presently simulated flight modes (in ISKED array).

Flight Mode	Code	INFO1	INFO2
VOR radial	1	VOR identification	Desired radial (deg) + toward - away
To non-directional beacon	2	NDB identification	-
VECTOR	3	Heading (deg)	-
Turn to new heading	4	Final heading (deg)	Direction 1 left 2 right
DME radius	5	DME identification	$\pm$ Desired distance (n.mi.)
Runway ILS	6	ILS identification	Orientation (deg)
Wait	7	-	-
Standard hold	8	-	-
Standard turn	9	Direction 1 left 2 right	-

Table 2. Presently simulated flight leg objectives (in ISKED array).

Leg Objective Mode	Code	INFO3	INFO4
VOR radial intercept	1	VOR identification	Degrees
Time	2	1 2	-- Time of day -- elapsed (sec)
Distance from initial leg point	3	Distance (n.mi.)	-
Altitude	4	Desired altitude (ft)	-
Fix	5	FIX identification	-
Speed	6	Desired speed (kts)	-
DME radius intercept	7	DME identification	Distance + toward - away
Continue until clearance received	8	-	-
Proceed immediately to new control point	9	-	-
Runway touchdown	10	Runway Fix identification	Distance (n.mi.)

TABLE 3

Altitude and Speed Pointers (in ISKED array)

Altitude Code	Definition
1-9	Points to desired altitude values for this leg listed in the "Scheduled Altitude" data input array (ALTSKD)
31	Indicates that desired altitude value will be found in associated "Altitude Option" data input array (ALTOPT)
0	Maintain altitude indicated by previous segment

Speed Code	Definition
0	Continue at initial speed
1-8	Points to desired speed value for this leg in associated aircraft speed performance data input array (IPERFS)

Flight Schedule Array (ISKED)

No.	Mode	INFO1	INFO2	Pointers Alt.	Pointers Spd.	Objective Mode	INFO3	INFO4	Length of Segment (ISDST)
1	1	3	12	31	0	3	9	-	9.0
2	1	3	12	31	0	5	3		8.2
3	3	220	-	31	0	1	8	103	16.4
4	3	270	-	7	7	2	2	600	27.7

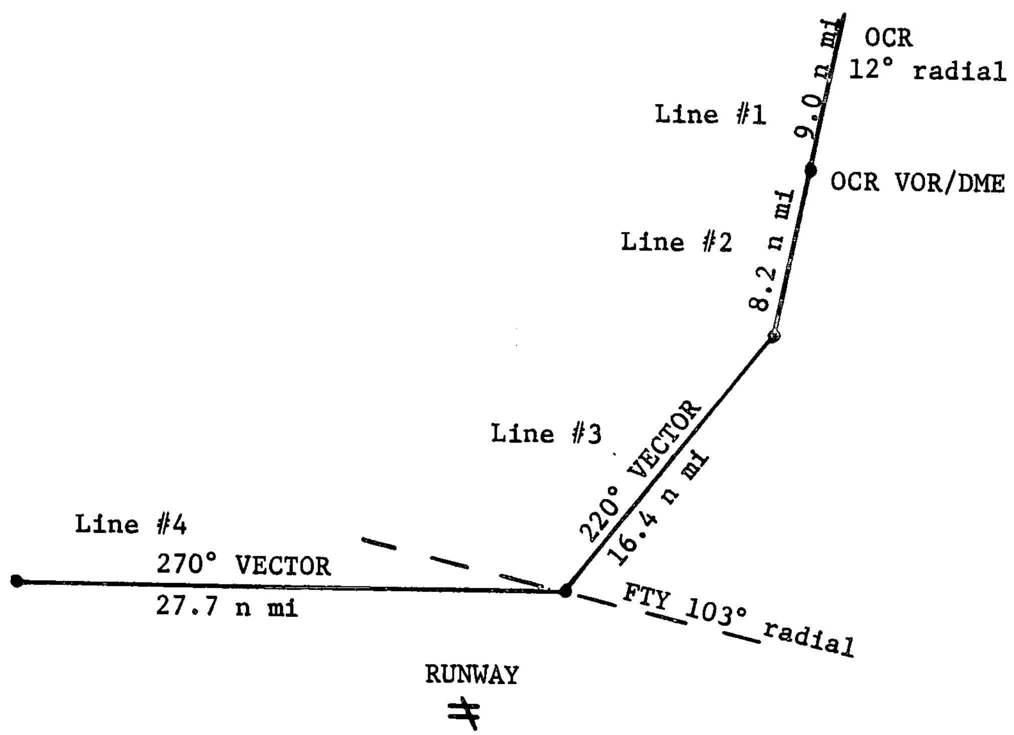


Fig. 5. Correlation of flight schedule data and route definition on map.

can be interpreted as follows:

Line 1 Fly toward OCR VOR on 12° radial until 9 n mi from present fix. Descend to controller-specified altitude. Maintain speed unless otherwise instructed.

Line 2 Fly toward OCR VOR on 12° radial until OCR is reached. Continue descent to controller-specified altitude. Maintain speed unless otherwise instructed.

Line 3 Fly 220° vector to intercept FTY 103° radial at controller-specified altitude. Maintain speed unless otherwise instructed.

Line 4 Fly 270° vector for 10 min (600 sec) at 6000 ft (6000 ft is specified as item #7 in the Altitude Schedule array, Appendix B, page 144). Slow to approach speed (speed code 7 identifies the approach speed in the Aircraft Performance Tables, Appendix B, page 124. **Aircraft will receive sequencing or** base leg vector instructions while on this leg).

Proceeding as described above, the entire set of nominal and alternate routes is defined in the Flight Schedule data input array, with each leg of flight corresponding to a line in the array. Simultaneously, the Scheduled Altitude data input array is filled out (Appendix B, page 151).

It should be noted that the scheduled flight plan may be overridden at any time by simulated controller action which may introduce a speed change, a standard hold, an emergency 360° turn or other path changes. These controller actions are discussed in the following section.

E. Simulated Controller Actions

The controller function is modeled in the simulator through the use of a series of conflict checkpoints (CHKPT) and controller action points (CNTRA) located at strategic points along the routes shown in Fig. 3. Detailed documentation of the tables and arrays of data used in conjunction with these points and the program logic used to implement the above procedures are given in the appendices. Here, an overall description is provided of how the system of CHKPTs and CNTRAs are utilized to simulate controller action at the Atlanta terminal.

CHKPTs are points on the routes where ETA information is maintained and at which conflict detection and resolution are effected. In essence, they demarcate portions of the flight path for which clearance needs to be

obtained. Hence they are generally located at intersections of paths and at critical points such as the ILS gate and the runway. CNTRAs are points on the routes at which the controller decision process of conflict detection and resolution is initiated. Each CNTRA has a particular CHKPT associated with it at which conflict detection and resolution are effected. Of course, a CNTRA is placed several minutes' flying time ahead of its associated CHKPT to allow time to resolve potential conflicts.

In the exercise of his routine functions, the human controller is faced with a wide variety of traffic situations. The various situations call for different types of controller decisions or actions, and the decisions have to be taken under differing sets of options open to him. To model this aspect of the ATC function, several different types of controller actions are provided for in the simulator, as listed in Table 4.

Table 4. Controller actions.

Code	Type Action
1	Arrival clearance - action to be taken on an arriving aircraft being "offered" at the terminal perimeter
2	Departure clearance - action to be taken on a departing aircraft being "offered" at a runway apron
3	General clearance - action to be taken on any airborne aircraft approaching its clearance limit
4	Sequencing - action to be taken on arrivals to sequence them for final approach
5	No conflict check necessary
6	Pop-up - action to be taken on any aircraft suddenly appearing at a random point in terminal airspace
7	Re-check aircraft for clearance
8	Hand-off to another controller (not yet programmed)
9	Register aircraft in "possible altitude conflict" array
10	Remove aircraft from "possible altitude conflict" array
11	Re-check pop-up aircraft for clearance

Each CNTRA point is assigned one of the preceding type numbers, depending on what sort of controller action needs to be initiated at the point.

Controller Action Input Data

To provide the input data necessary to specify the relatively complex controller actions that take place, several data input sheets are used. The input data arrays associated with the control function are listed in Table 5. The associated data sheets are given in Appendix B, pages 132-157.

These data arrays are nested and related by a system of pointers that indicate to the program where the appropriate input data will be found. A pointer in a given column of a data array indicates which item of a different array is referenced. The pointer system relating the input data arrays is diagrammed in Fig. 6.

The use of CNTRAs and CHKPTs and associated data may be illustrated by a specific example applied to Route No. 7 of the Atlanta route structure. A sketch of this route is shown in Fig. 7. Figure 8 shows some of the corresponding data entries in the input data sheets. Referring to Fig. 7, note that CNTRA 9 is at the perimeter on Route 7, and this indicates that some controller action is immediately called for. In Fig. 8, it may be seen that CNTRA 9 is specified as the starting control action for Route No. 7. Also, it may be seen from Fig. 8 that CNTRA 9 is a type 1 point, that is, one that checks for initial arrival clearance. Further, it must check for conflict up to CHKPT 5, which is the DALLAS intersection. Thus, CNTRA 9 has the function of insuring separation from previous arrivals approaching DALLAS on Routes 7 and 8. The CHKPT 5 data also indicate that the separation to be maintained at CHKPT 5 is 5 n mi and that this separation should be maintained for 14 mi behind the CHKPT. The associated ETA array is No. 5, and its coordinates are provided in the ETA array data sheet.

Control action 9 also references Option 10, which indicates the options to be used to resolve conflicts that are noted at CHKPT 5. Referring to line 10 in the Option Array of Fig. 8, we see that no altitude options are specified; if the aircraft is a jet it descends to 11,000 ft, and if it is a prop it descends to 6000 ft (at the rates of descent specified in the Aircraft Performance Array). The desired speed pointer points to the terminal speed, which is specified in the associated Aircraft Performance Array as the terminal speed for the aircraft entering. The nominal path

Table 5. Input data arrays associated with controller actions.

1. Controller Action Array (ICNTRA) - Identifies type of action to take place, associated checkpoint data array, option data array, the next controller action to take place, and the responsible controller.
2. Conflict Checkpoint Array (ICHKPT) - Specifies separation desired at the point and distances ahead and behind that separation should be maintained. Identifies associated ETA checkpoints and airspace conflict regions.
3. ETA Checkpoint Coordinate Array (IETCOOR) - Identifies the ETA files and specifies their x-y coordinates.
4. Controller Option Array (IOPTN) - Specifies options available to resolve conflicts. Identifies associated altitude option data array, desired speed, path number array and divergent route array.
5. Altitude Option Array (ALTOPT) - Specifies nominal and optional altitudes for both prop and jet.
6. Path Number Array (NNPATH) - Specifies number of segments in a given path, identifies the path, points to optional path, and associates the path with flight legs in the flight schedule array.
7. Divergent Route Array (IDROUTE) - Identifies checkpoints, options, and control actions for a possible divergent route.
8. Limit A/C Array (LIMAC) - Provides for changing separation standards at a specified ETA checkpoint as a function of the number of aircraft in a specified ETA array or queue.
9. Starting Route Array (ISTARTA) - Specifies the initial controller action that occurs on each route.
10. Speed Deviations (SPDSTD) - Specifies allowable deviations on each speed type for use in the speed control option logic.
11. Several miscellaneous inputs such as arrival/departure separation standards are provided as shown and defined in Appendix B, page 153.

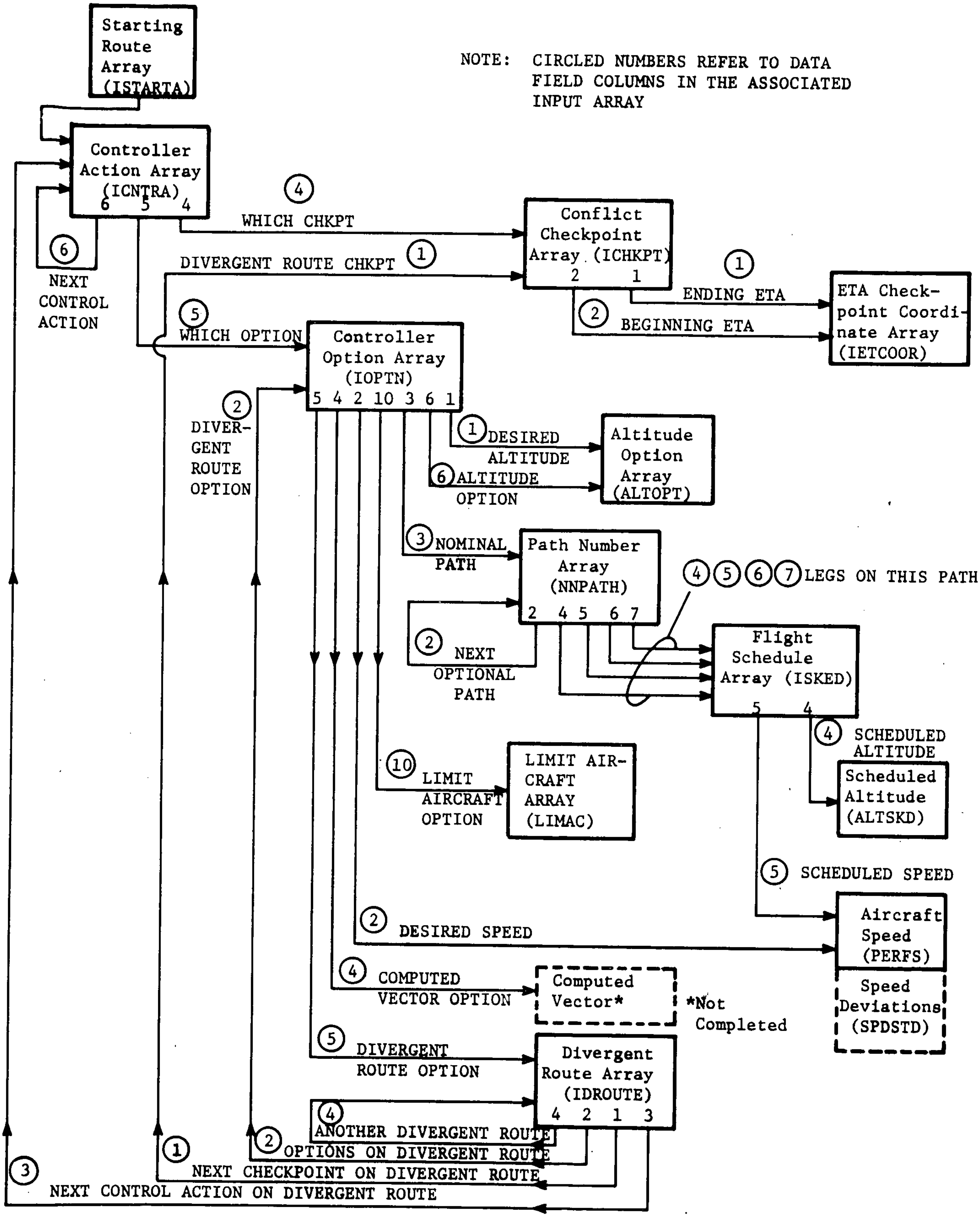


Fig. 6. Diagram of relationships between input data arrays as defined by pointers in the arrays.

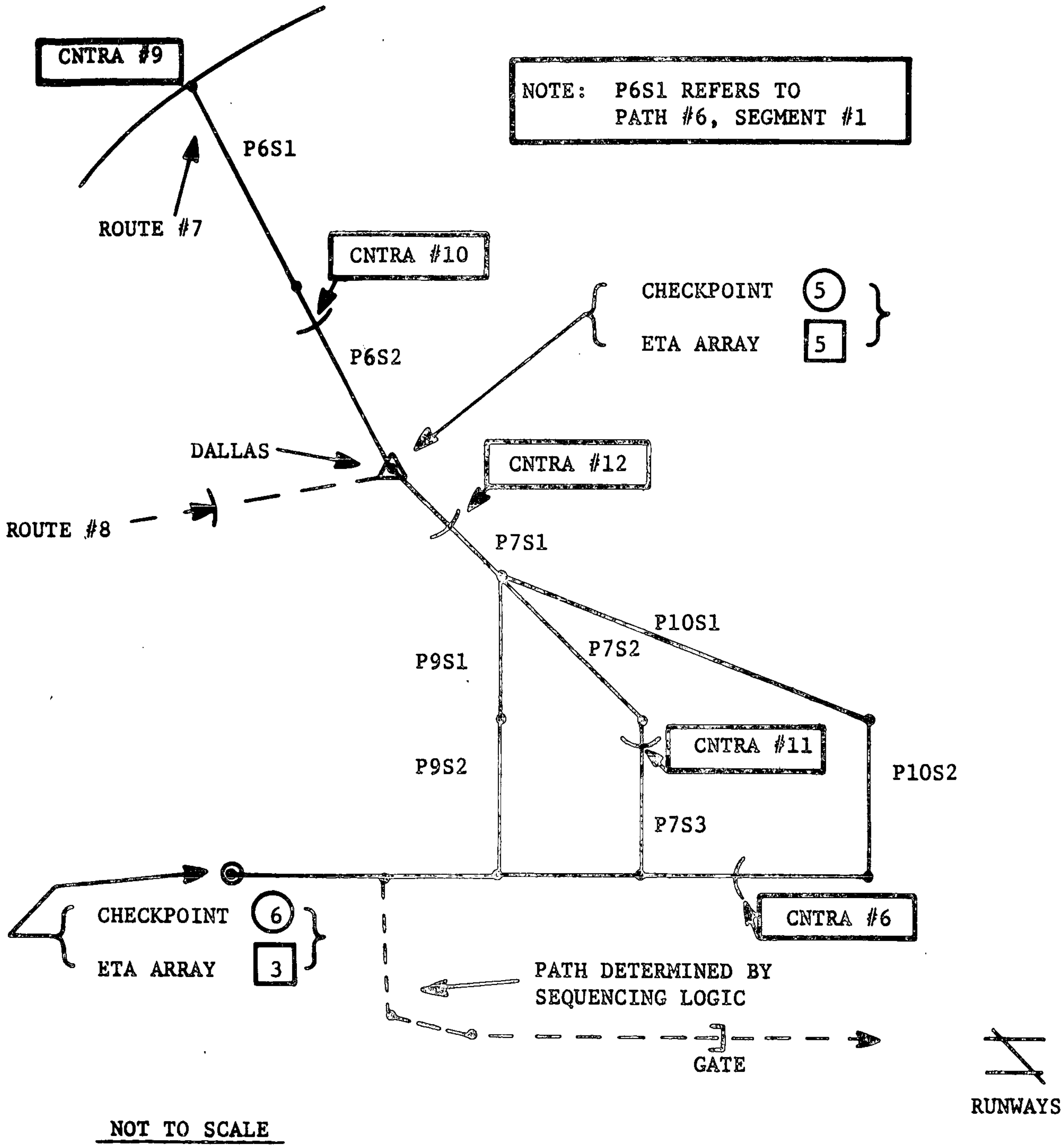


Fig. 7. Sketch of Route #7 of the Atlanta Terminal Area route structure illustrating the use of control action points and checkpoints.

pointer (6) references two segments (11 and 12 in the Flight Schedule Array) which indicates, as discussed previously, how the aircraft flies path 6. Control Option 10 indicates that the only option available for resolving a conflict at CHKPT 5 is a speed change. If the conflict cannot be resolved by a speed change, the aircraft is held in the enroute queue and is, in effect, not accepted at this time. If no conflict exists, or if the conflict has been resolved by a speed change, the aircraft will proceed along path 6 in accordance with the aircraft dynamics and the algorithms defining the flight mode.

As the aircraft passes CNTRA 10, further control action is triggered. Referring to Fig. 8, the data sheet indicates that CNTRA 10 is a type 3 (general clearance) point, and the associated checkpoint is CHKPT 6. Referring to Fig. 7, we see that two divergent routes are available (path 10 and path 9). The nominal path is path 7. Options specified as available to resolve a conflict at CHKPT 6 are specified in Controller Option 5 and include the divergent routes, a possible standard hold at DALLAS, and a possible emergency hold at any point along the path.

When the aircraft is finally cleared to CHKPT 6 by resolution of any conflicts, the aircraft will next encounter either CNTRA 11, 12 or 6. Each of these Control Action points is identified (see Appendix B) as a type 4 (sequencing) checkpoint. Passing one of these Control Action points triggers the sequencing logic, which determines the time to turn off the downwind leg and properly sequences the aircraft in a conflict-free manner to the runway touchdown point.

This general procedure is followed on all routes desired in the terminal area. For the route structure defined by the data sheets in Appendix B, the Control Action points, conflict checkpoints and associated data are indicated on the maps given in Fig. 9 (arrivals) and Fig. 10 (departures).

Note that on these maps, regions where arrival routes and departure routes intersect are designated as airspace conflict regions (ASPACE). These regions are defined by the special Control Action points types 9 and 10, which are placed on both arrival routes. Arrivals passing a type 9 point are registered in a special file against which departure aircraft wishing to climb through are checked for possible conflict at the arrival altitudes in use. The type 9 action point registers the arrivals in a specified file, and the type 10 action point removes them from the file.

Initial Controller Action Subscripts for Each Route (ISTARTA(I))

Route No. I	Controller Action Subscript (ICNTRA)
7	9

Controller Action Array (ICNTRA - Packed Array)

No.	Type	x North (n mi)	y East (n mi)	Checkpoint No. (ICHKPT)	Controller option (IOPTN) or Airspace pointer if type 9 or 10 (ASPACE)	Next Controller Action (ICNTRA)	Controller No.	Distance to specified checkpoint from preceding checkpoint (n.mi.)
9	1	28.4	-28.2	5	10	10	1	16.4
10	3	21.2	-23.1	6	5	11	1	23.0

Conflict Checkpoint Array (ICHKPT - Packed Array)

No.	No. of ending ETA Point	No. of Beginning ETA Point	Separation (n.mi.)	Maintain Separation Ahead (n.mi.)	Maintain Separation Behind (n.mi.)	Subscript for Airspace Conflict (ASPACE)
5	5	0	5	0	14	0

Controller Option Array (IOPTN - Packed Array)

No.	Pointer to Desired Altitude Array (ALTOPT)	Desired Speed Pointer (IPERFS)	Nominal Path Pointer (NNPATH)	Computed Vector Option (0=no, 1=yes)	Divergent Route Pointer (IDROUTE)	Altitude Option (0=no, 1=yes)	Speed Change Option (0=no, 1=yes)	Standard Hold Option (0=no, 1=yes)	Emergency Hold Option (0=no, 1=yes)
10	7	6	6	0	0	0	1	0	0
5	2	7	7	0	1	0	0	1	1

ETA Checkpoint Coordinates (IETCOOR - Packed Array)

ETA No.	X Coord. North (n mi)	Y Coord. East (n mi)
5	15.00	-18.70

Altitude Option Array (ALTOPT(I,J,K)) (ft)

Altitude Option No. (I)	Pointer From Option Array (J): 7
1	6000 / 11000
2	0 / 0
3	0 / 0

K = 1: prop
K = 2: jet

prop / jet

AIRCRAFT PERFORMANCE

(IPERFS - Packed Array)

Speeds (Kts)

1 Take-off	2 Climb to 10,000 ft	3 Climb to 20,000 ft	4 Cruising	5 Transition	6 Terminal	7 Approach	8 Final
					200		
					200		
					175		

Path Number Array (NNPATH - Packed Array)

Path Subscript from IOPTN	No. of Segments	Pointer to Next Optional Path	Path No.	Subscripts to Schedule Array (ISKED): 1st Segment	2nd Segment	3rd Segment	4th Segment
6	2	0	6	11	12	0	0

Divergent Route Array (IDROUTE - Packed Array)

No.	Pointer to Checkpoint Array (ICHKPT)	Pointer to Option Array (IOPTN)	Pointer to Controller Action Array (ICNTRA)	Pointer to Another Divergent Route (IDROUTE)	Distance to specified checkpoint from preceding checkpoint (n.mi.)
1	6	6	12	2	14.9
2	6	7	6	0	35.6

Flight Schedule Array (ISKED - Packed Array)

No.	MODE	INFO1	INFO2	Pointers ALT	Pointers SPD	CP MODE	INFO3	INFO4	Length of Seg. (ISDST) (n mi)
11	1	1	325	31	0	3	9.0	—	9.0
12	1	1	325	31	0	1	3.0	264	7.4

Fig. 8. Entries in the input data sheets illustrating the correlation of the input data with the route structure of Fig. 7.

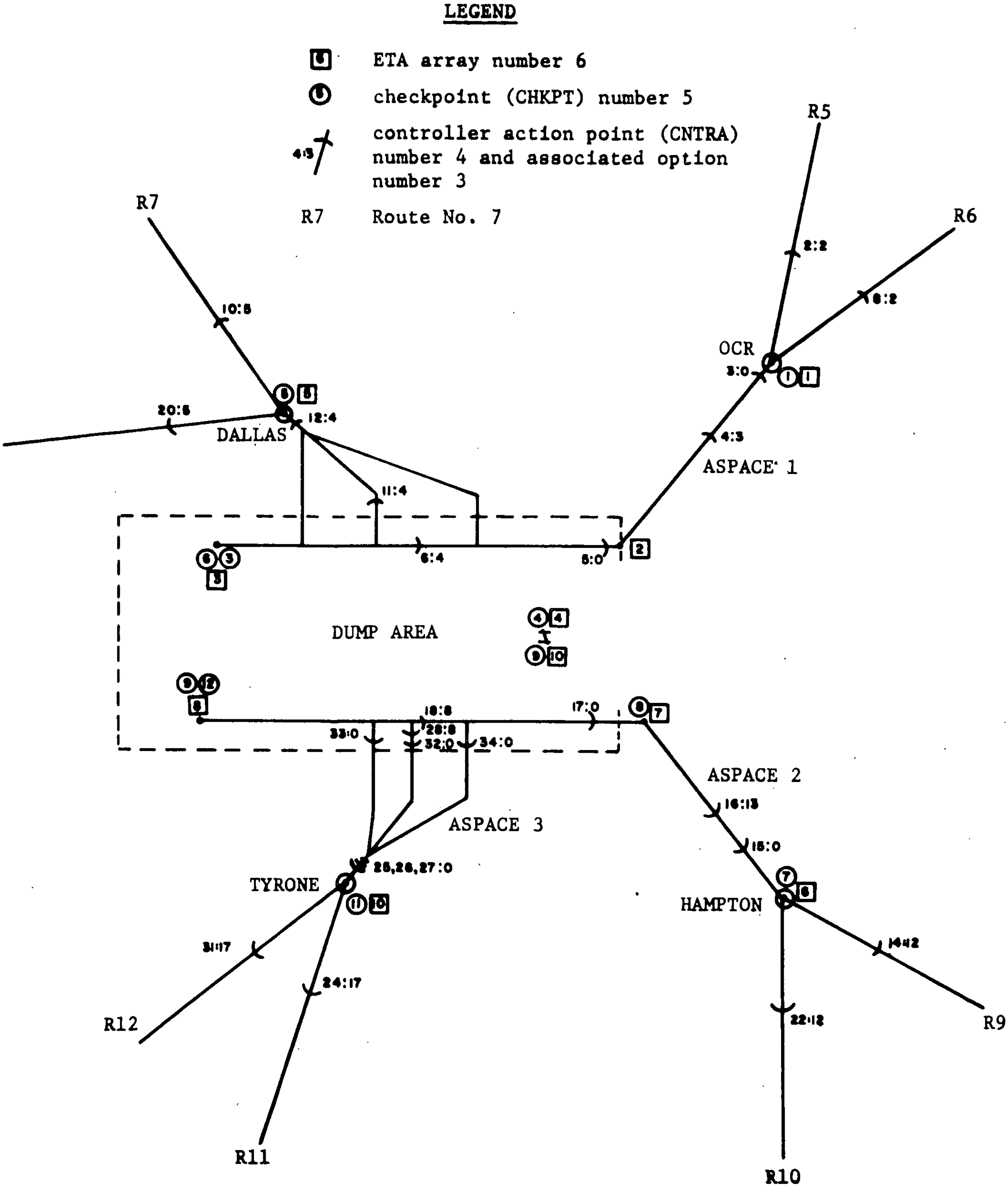


Fig. 9. Sketch of Atlanta arrival routes represented by the data given in Appendix B, showing the conflict checkpoints and the controller action points along the various routes.

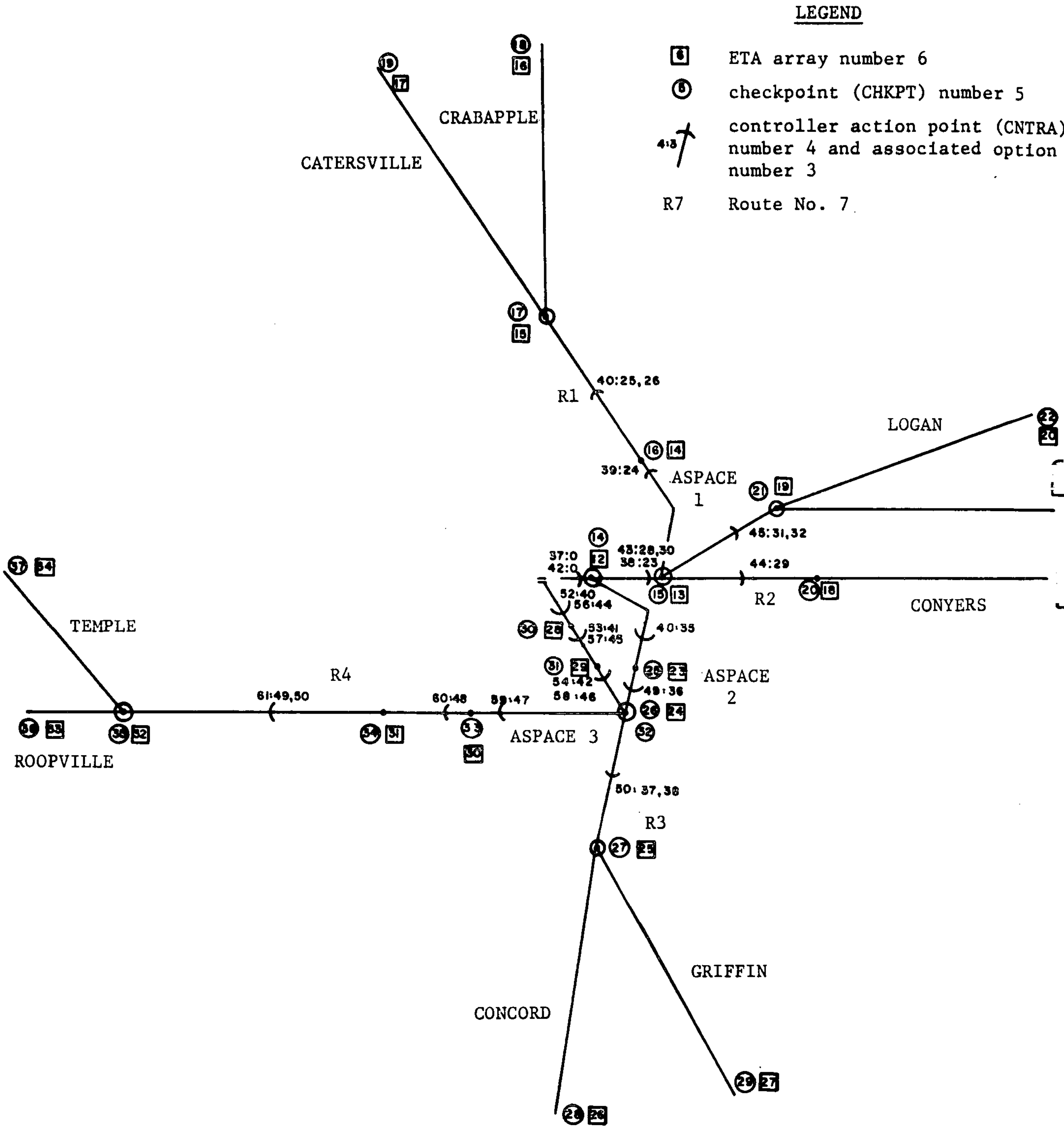


Fig. 10. Sketch of Atlanta departure routes represented by the data given in Appendix B, showing the conflict checkpoints and the controller action points along the various routes.

It should also be recalled that, as in reality, the aircraft do not follow the specified routes exactly. Deviations occur because of specified random navigation and surveillance errors and because, while the routes are approximated by straight-line segments in most cases, the actual paths are determined from aircraft dynamic equations and have differing turn radii for different aircraft types and speeds.

F. Final Approach Sequencing

Final approach sequencing (FAS) is the procedure used by the tower controllers to direct arrival aircraft within the FAS dump area to the runway touchdown point. The FAS dump area at Atlanta is a rectangular area extending some 30 n mi on the approach side of the runways. For runway 9L the north boundary is the FTY 90/270 radial and for runway 9R the south boundary is the ATL 90/270 radial. Altitudes in this controlled area are generally restricted to 6,000 ft MSL. Specifically, the FAS controllers have the task of maintaining separation between aircraft within the dump area while directing them along the downwind leg, the base leg, and onto a localizer intercept course. Normally, VFR aircraft will be controlled and directed along the localizer course, past the ILS gate, and their progress monitored until touchdown. IFR aircraft are directed onto a localizer intercept course and then monitored as they proceed along the localizer course down the glide slope to touchdown.

The FAS controller must insure aircraft safety while maintaining as high a landing rate as possible so that efficient and expeditious movement through the dump area is afforded. The terminal area simulation model employs an algorithm to simulate the control provided by a human controller within certain well-defined constraints. Continuous separation of aircraft is the primary goal.

Generally, all aircraft are directed into the dump area by passing over an inner fix point. A sequencing queue containing at most three aircraft is established to provide for calculation of the necessary control times to move an aircraft from an inner fix to touchdown. As each aircraft crosses an inner fix it is inserted into this sequencing queue. Of all the previously sequenced aircraft in the dump area which have not reached the ILS gate, the sequencing queue contains up to two aircraft with the latest scheduled ILS gate times.

Upon entry into the sequencing queue, a time to turn off the

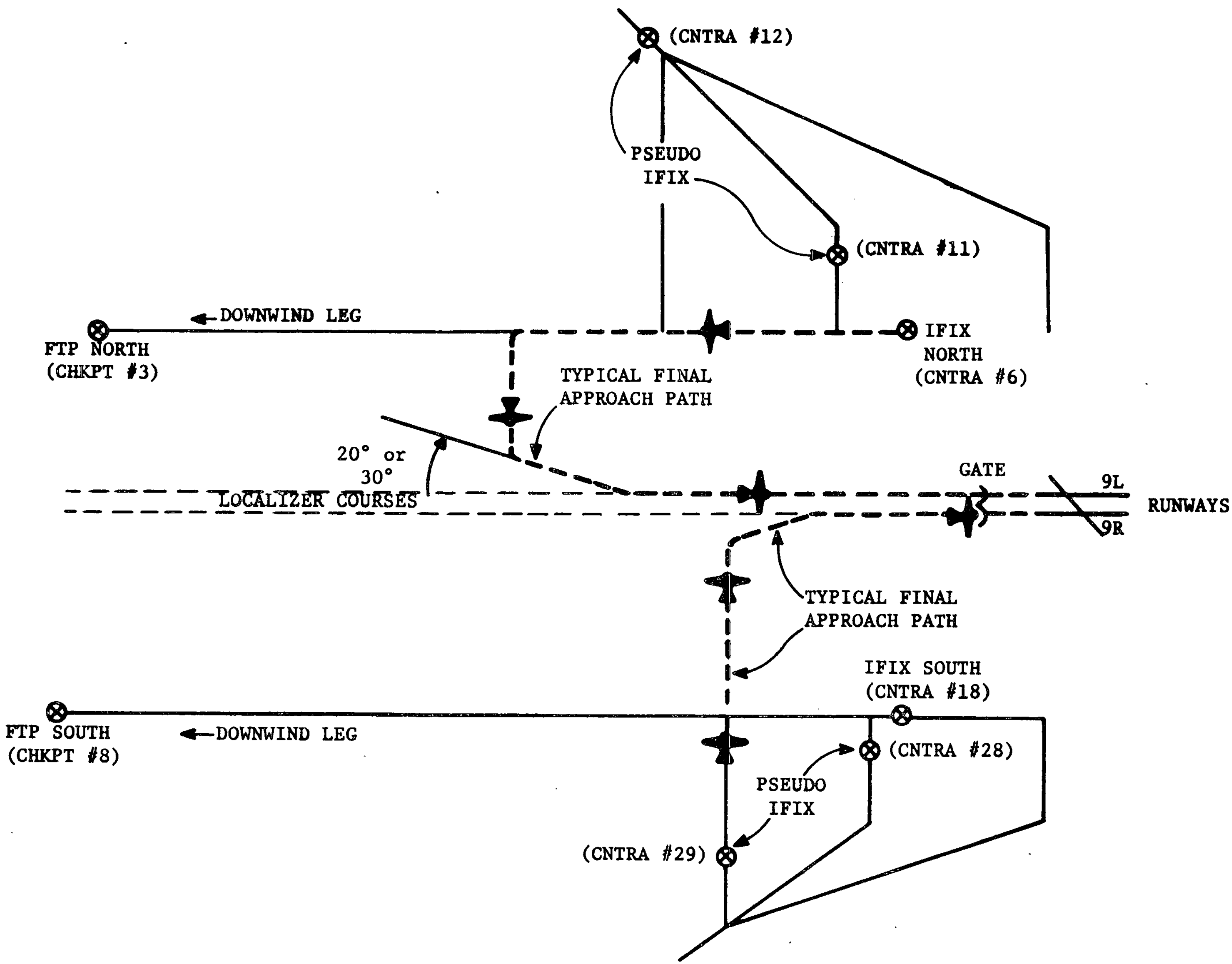


Fig. 11. Typical north and south arrival final sequencing paths illustrating approach paths and use of control points in sequencing technique.

downwind leg onto the base leg vector is calculated (TOFF) along with a time to turn onto the localizer intercept vector (TONN). These times are based on providing proper longitudinal separation at the ILS gate using an estimated time of arrival at the gate. The basic structure of the approach paths along with refinements to the algorithm assure separation between aircraft throughout the dump area. Typical paths and the associated control points are illustrated in Fig. 11.

The parallel runways are fed arrival aircraft by partitioning the dump area into north and south sections such that all arrivals into the north portion of the terminal area are sequenced in the north portion of the dump area, while southern arrivals use the south portion of the dump area. Since the two runways 9L and 9R do not have enough lateral separation to provide for independent operation, it is necessary for the algorithm to treat the runways as dependent.

With the Atlanta route structure arrival aircraft using runways 9L and 9R, two inner fixes (IFIX) are defined. The north inner fix for runway 9L is defined on the downwind leg west of the ILS gate and is chosen as the earliest possible turn-off point (EPTP). No aircraft is allowed to turn off the downwind leg before reaching the inner fix; thus, the time of crossing the inner fix is the time at which each aircraft enters the sequencing queue. It can be noted in Fig. 9 that there are four paths which feed into the dump area from the north arrival sector. One is for NORCROSS arrivals while the other three represent the nominal and alternate routes for DALLAS arrivals. A point on the downwind leg termed the final turn-off point (FTP) is within the dump area and is the end of the downwind leg for sequencing purposes. Two of the feeder paths are such that an aircraft arriving over DALLAS might not actually fly over IFIX. For this reason, a pseudo-inner fix is defined on each of these two paths. These pseudo-inner fix points are defined such that the time required to move from the pseudo-inner fix (IFX1 or IFX2) to the FTP is the same as if the aircraft actually passed over IFIX. The time the aircraft reaches the pseudo-inner fix then becomes the time when the aircraft enters the sequencing queue. For these aircraft, the earliest possible turn-off point located on the downwind leg does not correspond to the sequencing point. For this reason, a special provision is made to determine the EPTP for these aircraft.

The south arrival situation is much the same as in the north. For

aircraft arriving over HAMPTON, the south IFIX is the sequencing point. For TYRONE arrivals, two of the three alternate feeder routes have pseudo-inner fixes as sequencing points. Again, the FTP on the downwind leg south is used to determine the time of entry into the sequencing queue. As each aircraft is entered into the sequencing queue, calculations of TOFF and TONN are made to provide for proper separation at the ILS gate. An aircraft entering the queue, termed the current aircraft, is first checked to see if it can be allowed to pass the aircraft already in the queue and scheduled to immediately precede the current aircraft at the ILS gate. The current aircraft may pass only one preceding aircraft and then only if certain criteria are met. It should be noted that one FAS queue handles both north and south arrivals for this dependent parallel runway situation.

To establish the inner fixes and pseudo-inner fixes, sequencing type controller action points (e.g., CNTRA's 11, 12 and 6 in Figs. 7 and 11) are used and are located such that they are equidistant from the ETA checkpoint (CHKPT 6 in Figs. 7 and 11). At the time an aircraft reaches one of these sequencing type controller action points it is entered into the sequencing queue. The use of the pseudo-inner fix points enables the sequencing algorithm to be used for general feeder path structures.

Input data required for specifying the information associated with the sequencing logic is defined and given in Appendix B, page 153.

G. Communications

Basic information required concerning the communications logic in the program includes specification of the particular controller having jurisdiction over particular segments of the path and the durations of the messages transmitted.

The controller responsibility areas, i.e., which controller is responsible for which action, are specified in the Controller Action Array. For example, in the Controller Action Array data listed in Fig. 8, Controller No. 1 is responsible for Control Action No. 9. Each controller action has the associated controller specified.

For each controller specified (representing each communications channel), messages are handled by setting up two message files in the computer. One of these files consists of future messages that are to be

sent at a calculated time and the other consists of current messages which are to be sent as soon as channel occupancy permits. For each message to be transmitted, an earliest transmit time (ETT) and a latest transmit time (LTT) are calculated. The message, when initially generated, is automatically placed in the future message file, and it is then transferred from the future message file to the current message file when the clock time equals the ETT. If the communication channel loading is such that a message cannot be transmitted by the latest transmit time, the indicated maneuver is carried out by the aircraft and a typewritten error message is generated to note that this situation existed. Each schedule change has associated with it the appropriate message phraseology.

For each class of message transmitted, the time required to transmit the message and to receive the pilot's acknowledgement is listed in the Controller Message Length Array. For the present Atlanta simulation, this array is given in Appendix B, page 151.

H. Navigation and Surveillance Errors

Errors associated with the navigation aids were previously discussed in Section IV.C. and appropriate data listed for each type of navigational aid in the Navigational Aid Array, Appendix B, page 126. Errors that reflect aircraft or pilot-dependent heading errors are provided as input to the program in the "Aircraft Type-Dependent Random Heading Error Array" (IACFPER), as shown in Appendix B, page 152. Each aircraft type has an associated type-dependent heading error specified.

Surveillance errors are specified separately as range, bearing and altitude standard deviations and are provided as input in the miscellaneous data input data sheet given in Appendix B, page 153.

I. Miscellaneous Inputs

Various miscellaneous inputs are required to define runway coordinates, runway separation information, glide slope angles, scale factors, etc. These miscellaneous input parameters are defined in Appendix B, page 153. This appendix also lists the values currently being used for the Atlanta simulation.

Preceding page blank

V. INITIAL RESULTS FROM TEST RUNS

A. General Discussion

During the course of the development of the model, several versions of the model have been used for test runs. To distinguish these versions of the model, a nomenclature has been developed to distinguish between the various versions. This nomenclature is as follows:

Atlanta 1

This version of the model is essentially the one described in Ref. 1. The route structure represented is that described in Atlanta Tower Orders dated 5/15/69.

Atlanta 2

A version of the model using the Atlanta route structure which went into effect 11 November 1971. Model algorithms were essentially those used in the Atlanta 1 version.

Atlanta 3

Minor modifications were made to the Atlanta 2 version to improve the realism.

Atlanta 4

This is the current version of the model, containing significant changes and representing the current Atlanta route structure that went into effect 11 November 1971.

An initial set of test runs was conducted using the old Atlanta route structure (Atlanta 1 version of the model) in order to compare the model output data with actual radar traffic data taken at Atlanta in 1967. The Atlanta 1 version represented the route structure in effect at that time.

In this section, the model output data are discussed and the test runs comparing the model output data to the Atlanta radar traffic data are presented.

B. Model Output Data

Operating in the real-time mode at the computer complex at LRC, several types of output data are obtained. The simulation console CRT permits the observation of the traffic flow in the terminal area with

PRECEDING PAGE BLANK NOT FILMED

various types of CRT displays. Also, strip chart recordings are used to record messages sent over communications channels and various parameters of interest, such as time histories of the model queues. In addition, data printouts are obtained which contain the scan-by-scan outputs giving the position of each aircraft, messages sent, sequencing data, and other information.

In the batch mode of operation, the primary data output is a data tape containing scan-by-scan tracking information to be used as input to the statistical analysis program, which at present provides statistical data on the traffic flow. The batch mode of operation will also print this scan-by-scan tracking data if desired.

Examples of the data output are shown in Figs. 12 through 18. Fig. 12 is a plot of the CRT display taken from a test run at LRC on 18 August 1972. Provision has been made for changing the scale on this plot, as is shown in Figs. 13 and 14. In Fig. 12, the range circles are 10 mi apart, in Fig. 13 they are 5 mi apart, and in Fig. 14 they are 2½ mi apart.

The console operator may also select an aircraft relative display which displays the traffic in the terminal area relative to a selected aircraft. An example of this type of display is shown in Fig. 15, where the traffic display is relative to the aircraft in the center of the plot. Various scales may also be selected for the relative CRT display.

Examples of the strip chart recording outputs are shown in Figs. 16 and 17. The data shown on Fig. 16 provide a time history of the messages transmitted by the various simulated controllers and the number of future messages in each controller's message queue. In this particular example, 8 controller channels are being simulated. The strip chart shown in Fig. 17 is an example of the type of time history data obtained on the number of aircraft awaiting entry into the problem area and the number of departures awaiting departure clearance as well as information on aircraft in two selected ETA files and the total number of aircraft in the problem area.

Figure 18 shows a sample printout of the scan-by-scan data giving the position of each aircraft at four-second intervals as well as other supplementary information on the individual aircraft.

Current outputs of the statistical analysis program are listed in Table 6.

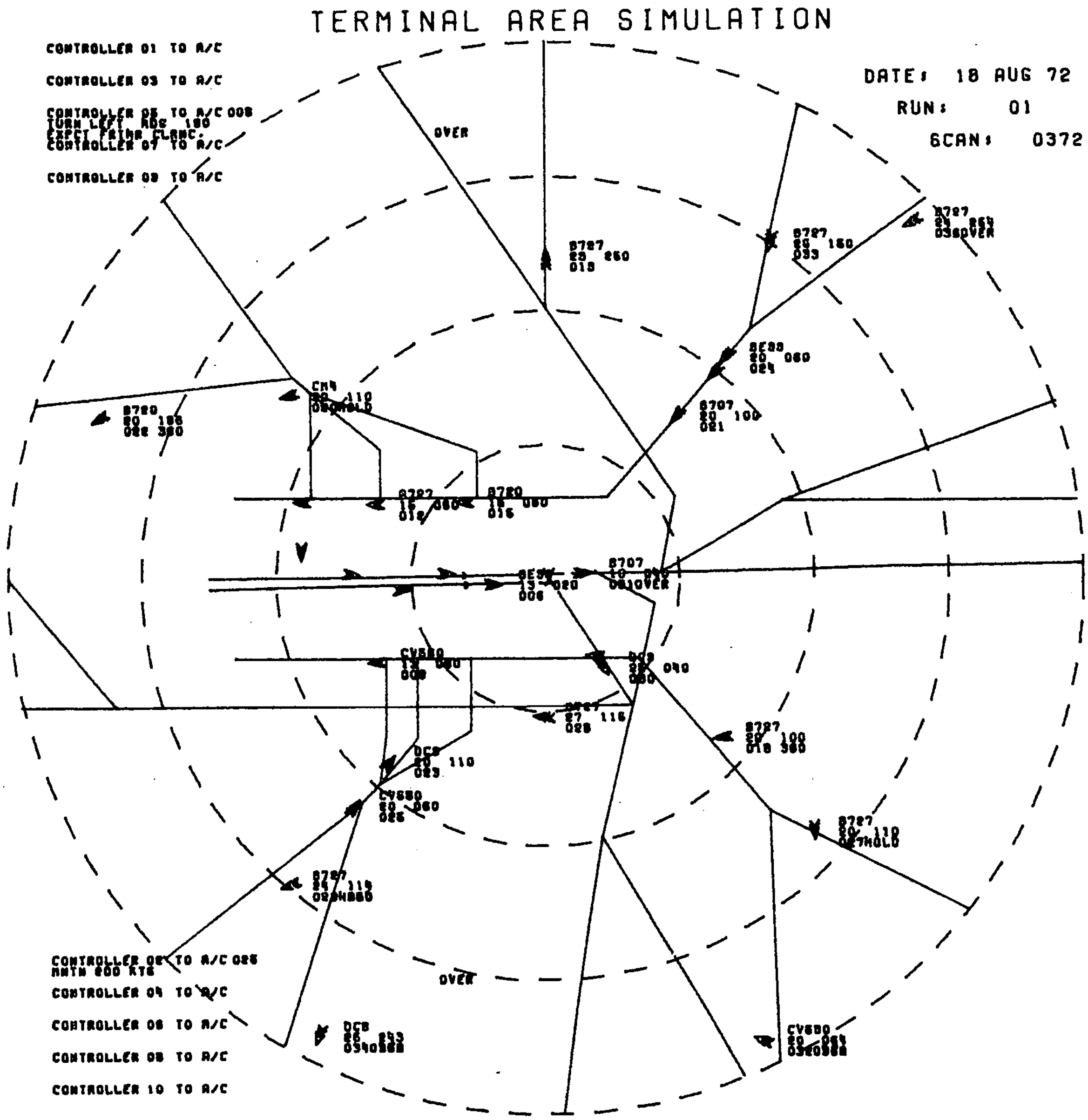


Fig. 12. The real-time CRT display. This plot is updated every four seconds in the real-time mode. The lines on the plot represent the nominal route structure. The dashed range circles are 10 n mi apart. This particular plot was made at scan 372 of a test run. Controller messages are shown in the upper and lower left-hand corners.

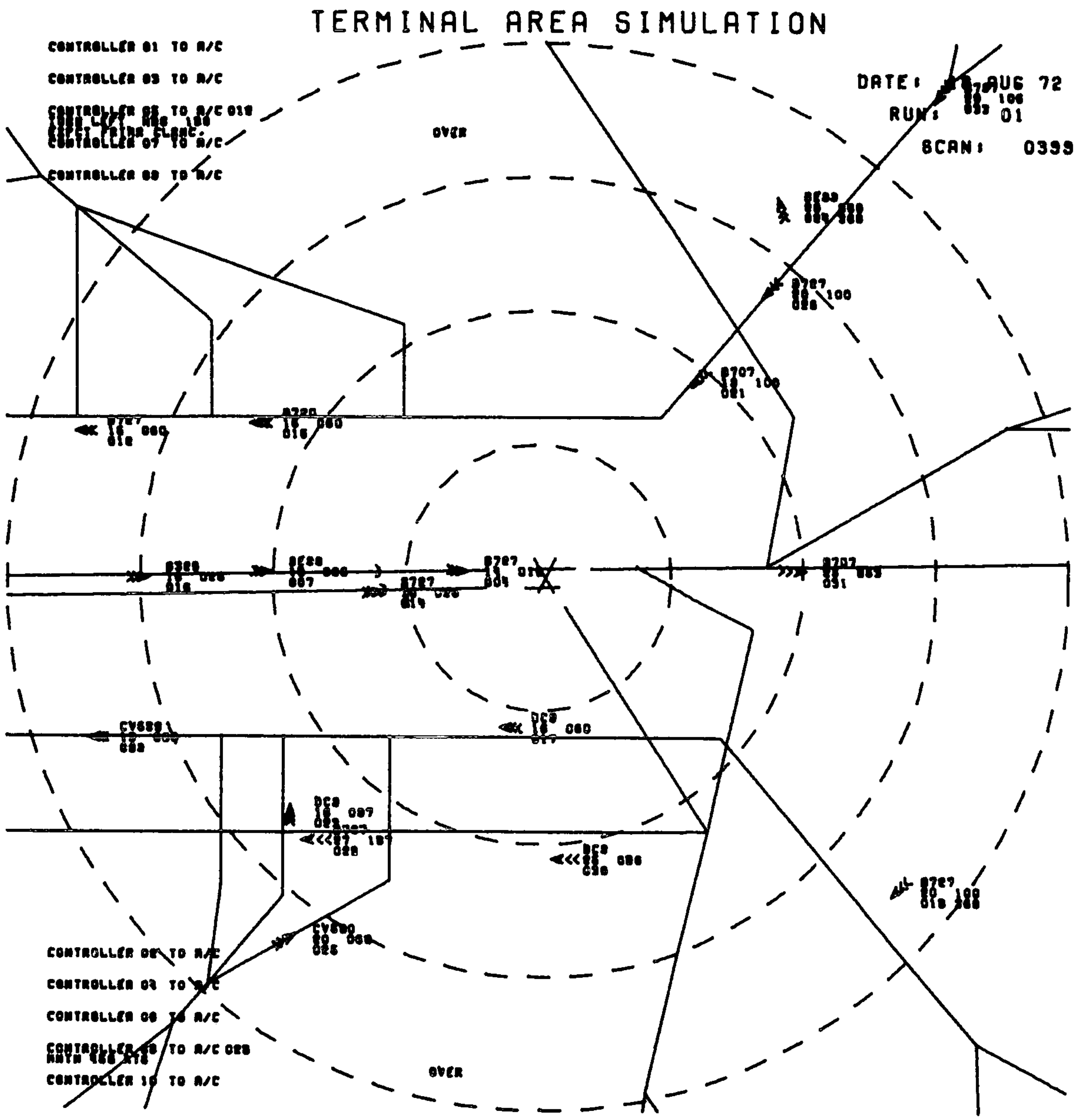


Fig. 13. The real-time CRT display. This plot is updated every four seconds in the real-time mode. The lines on the plot represent the nominal route structure. The dashed range circles are 5 n mi apart. This particular plot was made at scan 399 of a test run. Controller messages are indicated here in the upper and lower left-hand corners.

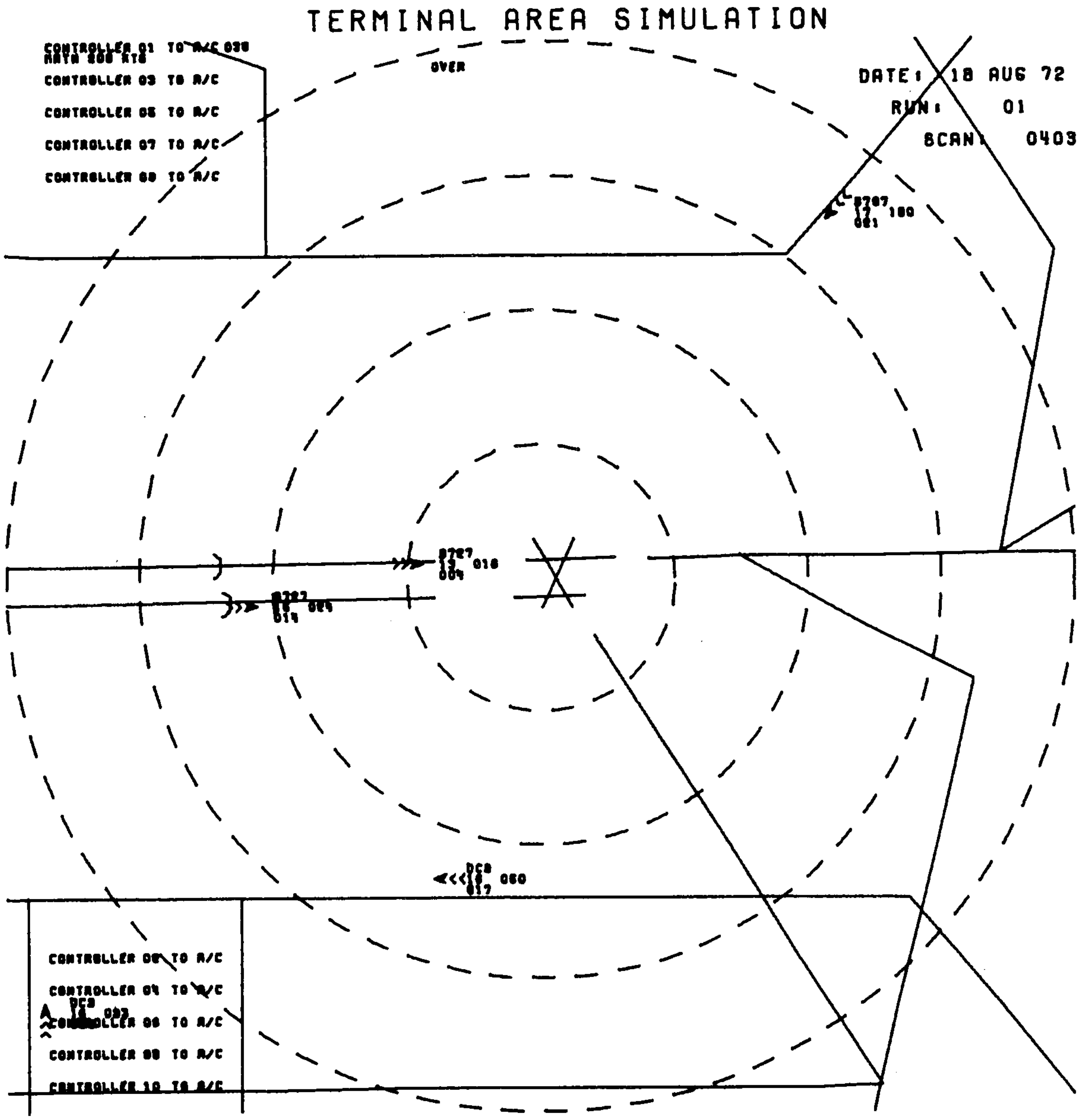


Fig. 14. The real-time CRT display. This plot is updated every four seconds in the real-time mode. The lines on the plot represent the nominal route structure. The dashed range circles are 2½ n mi apart. This particular plot was made at scan 403 of a test run. Controller messages are indicated here in the upper and lower left-hand corners.

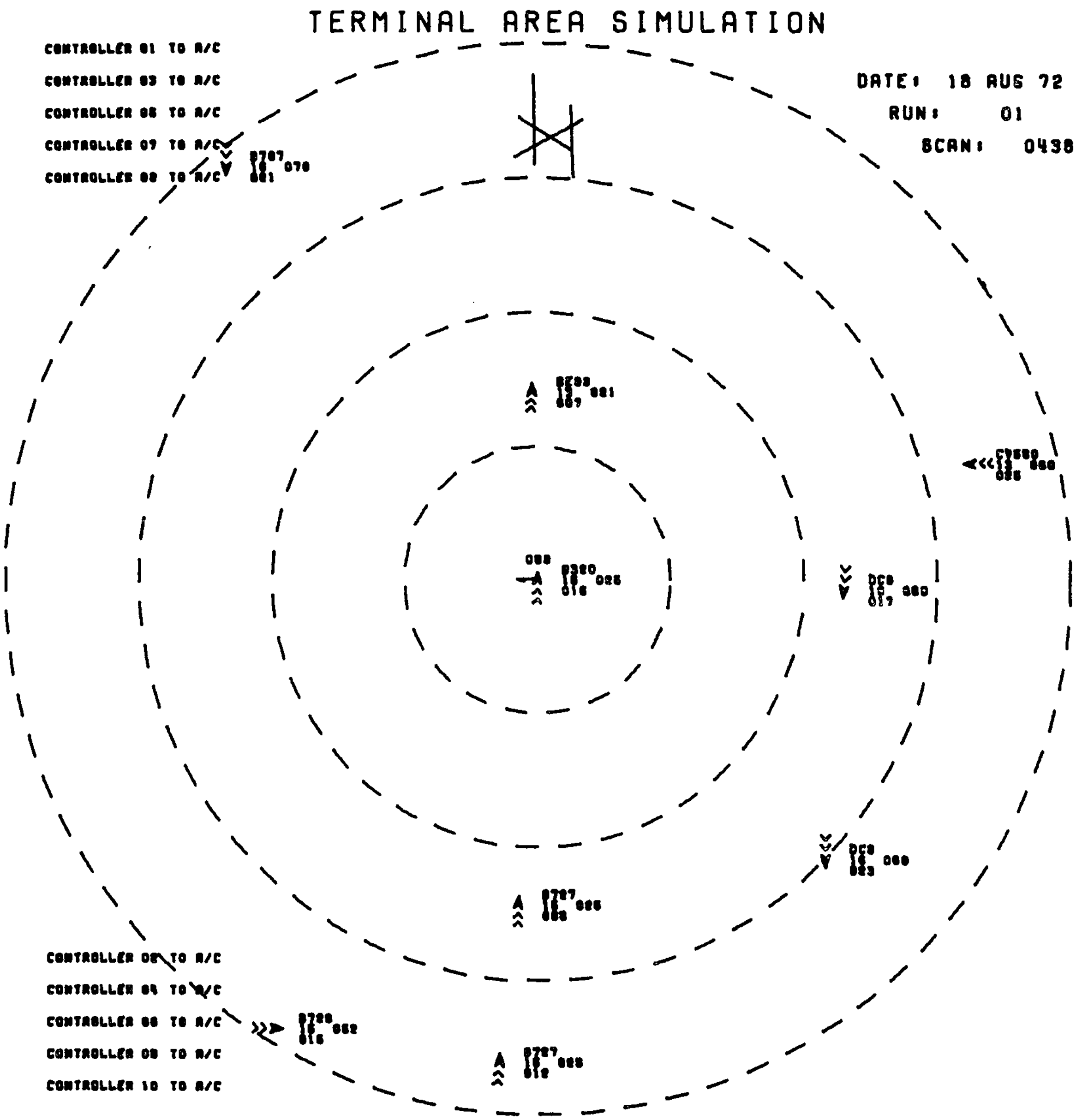


Fig. 15. The aircraft-relative real-time CRT display. Positions of the aircraft shown are relative to the aircraft in the center of the display. The plot is updated every four seconds in the real-time mode. The dashed range circles are 2 n mi apart. Scale on the plot is variable. Controller messages are indicated in the upper and lower left-hand corners.

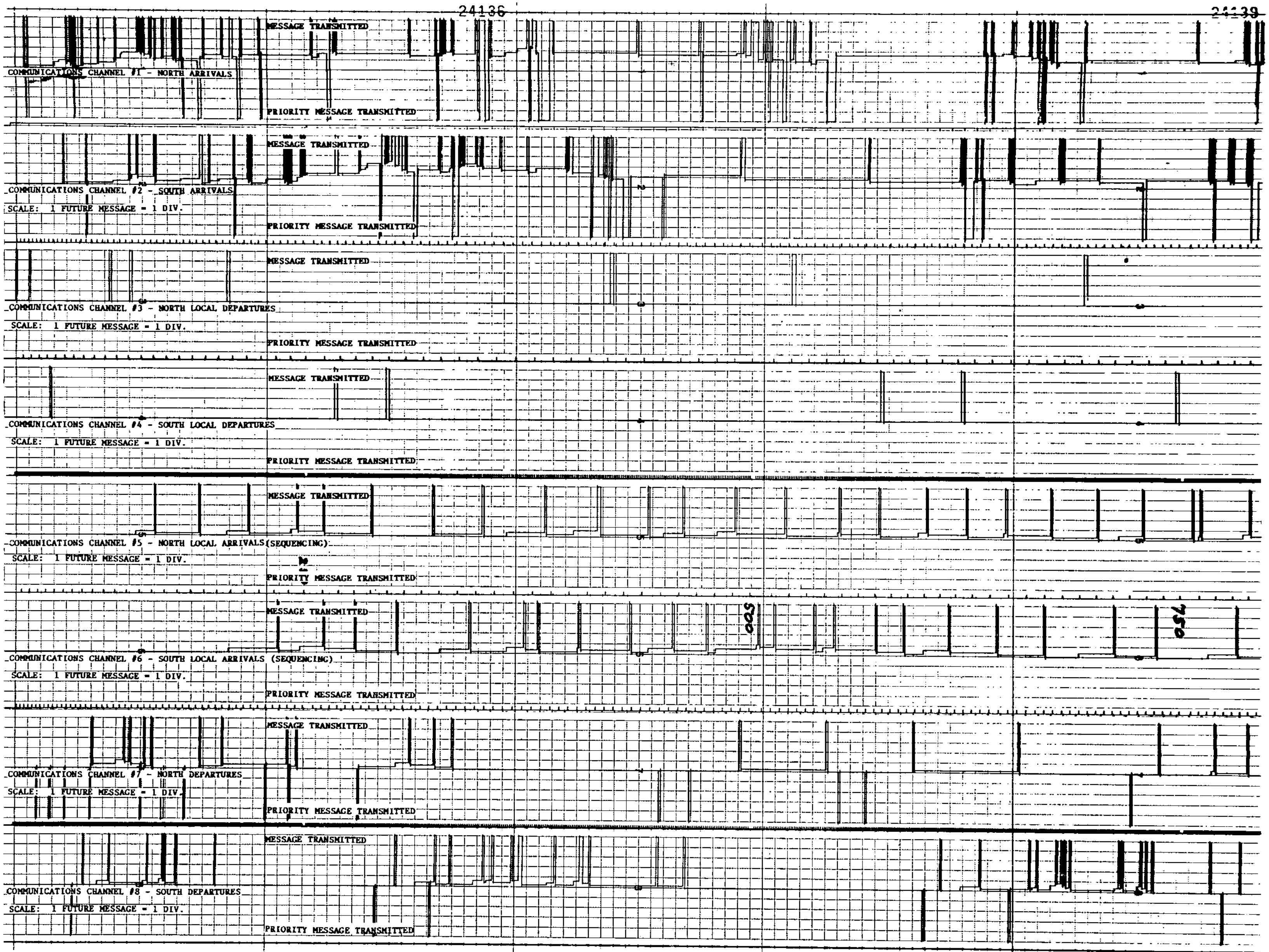


Fig. 16. Example of output data obtained on strip chart recording--communications channel information.

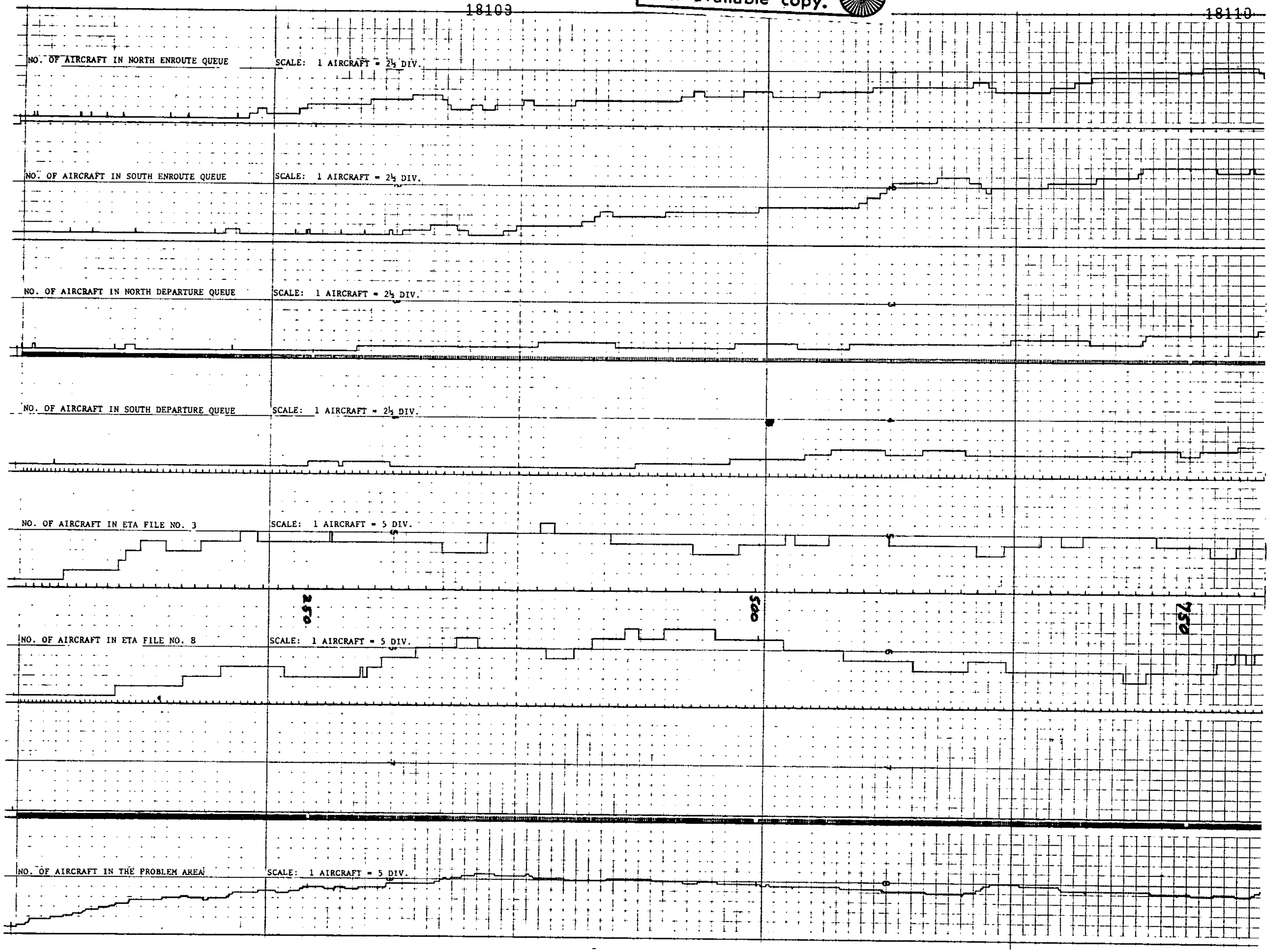


Fig. 17. Example of output data obtained on strip chart recording--time histories of various queues.

SCAN: 517 NO. A/C: 23 RUN: 4

	LATEST TRANS-MIT TIME	IDENTIFICATION	MESSAGE LENGTH IN SCANS	EARLIEST TRANSMIT TIME	NO. OF CLEARED SEGMENTS	TYPE	ETA SUBSCRIPT	INFO1	INFO2
MESSAGE ISSUED BY CONTROLLER 1	517	39	1	514	3	3	5	2	110

SCAN: 518 NO. A/C: 23 RUN: 4

AIRCRAFT IDENTIFICATION	X COOR. (n mi)	Y COOR. (n mi)	Z COOR. (ft)	SPEED (kts)	HEADING (deg)	ROUTE ASSIGNED	1=Dep.,2=Arr.	FLIGHT MODE / CONTROL POINT MODE	CURRENT FLIGHT LEG	X COOR. (n mi)	Y COOR. (n mi)	Z COOR. (ft)	PACKED CURRENT PATH INFORMATION	NO. OF SEGMENTS REMAINING	PACKED FUTURE PATH INFORMATION	NO. OF SEGMENTS REMAINING	CONTROLLER ACTION POINT
12	.45	-8.55	2500	150	92	6	2	601	4	.500	-8.564	2400	0000000000000000604	2	000000000000000000	0	0
15	3.57	-13.89	4500	150	180	6	2	1002	4	3.447	-13.886	4500	0000000000000000604	1	000000000000000000	0	0
16	.31	-.63	1136	128	90	8	2	610	3	.298	-.605	1036	0000000000000000604	1	000000000000000000	0	0
17	-.53	-5.02	2283	155	83	10	2	610	4	-.534	-4.977	2383	0000000000000002011	1	000000000000000000	0	0
18	-5.67	-12.22	6007	150	271	10	2	302	3	-5.574	-12.266	6007	0000000001205041610	1	000000000000002011	0	0
20	13.54	-15.43	11000	200	130	8	2	301	2	13.525	-16.549	10800	0000000001305034l203	3	000000000000000000	0	11
21	5.76	4.61	9500	160	245	6	2	302	3	5.734	4.556	9700	0000000001204014040 3	1	000000000000000000	0	5
25	-1.14	-11.97	4145	135	83	11	2	601	3	-1.190	-11.965	4045	0000000000000002011	2	000000000000000000	0	0
27	-5.72	-.13	6045	150	271	10	2	302	3	-5.775	-.169	6045	0000000001205041610	1	000000000000000000	0	18
29	-24.22	-21.92	15788	244	215	12	2	902	1	-24.379	-21.815	15788	0000000003113110001 2	3	000000000000000000	0	31
32	-6.07	3.92	6000	135	265	10	2	302	3	-6.112	3.904	5900	0000000000606054161 0	1	000000000000000000	0	17
34	-22.19	-14.44	11023	200	278	11	2	402	1	-22.156	-14.662	11023	0000000003213100001 2	2	000000000130610424 10	0	25
22	14.92	-21.55	11014	200	84	8	2	101	1	14.791	-21.524	10914	0000000003013040000 5	1	000000000130605012 03	0	6
24	19.31	15.46	6000	203	236	6	2	105	1	19.229	16.511	6100	0000000001306024000 1	1	000000000506010020 2	1	3
26	17.45	14.59	10343	200	216	5	2	301	2	17.314	14.504	10143	0000000001312010020 2	1	000000000000000000	0	4
35	-34.12	-19.75	20662	235	355	11	2	902	1	-34.226	-19.676	20762	0000000002613100001 2	3	000000000000000000	0	24
33	19.33	15.45	11000	216	193	5	2	105	1	19.475	15.355	11100	0000000003313004000 1	4	000000001312010020 2	0	3
36	26.53	26.47	23078	200	179	6	2	902	1	26.523	26.661	23178	0000000003413024000 1	3	000000000000000000	0	8
37	-10.67	10.26	6000	200	320	9	2	301	2	-10.734	10.211	6100	0000000000605060140 7	1	000000000060641610	0	17
31	.92	22.87	16933	270	89	2	1	308	4	1.022	22.881	17033	0000000001625174442 5	1	000000000000000000	0	0
38	13.90	-26.94	14119	200	121	8	2	902	1	13.869	-26.928	14019	0000000003513040000 5	6	000000001306034120 3	0	11
43	.40	6.77	3957	250	90	2	1	301	2	.390	6.848	3957	0000000000404144301 5	1	000000000000000000	0	43
39	25.58	-25.50	24472	217	168	7	2	902	1	25.744	-26.517	24372	0000000003413030000 5	3	000000000000000000	0	10

	LATEST TRANS-MIT TIME	IDENTIFICATION	MESSAGE LENGTH IN SCANS	EARLIEST TRANSMIT TIME	NO. OF CLEARED SEGMENTS	TYPE	ETA SUBSCRIPT	INFO1	INFO2
MESSAGE ISSUED BY CONTROLLER 1	532	22	1	517	4	0	3	0	0

SCAN: 519 NO. A/C: 23 RUN: 4

Fig. 18. - Sample printout of the scan-by-scan data.

Table 6. Current outputs of the statistical analysis program.

1. List of arrival and departure aircraft, giving:
   a. aircraft identification
   b. scheduled entry time
   c. actual entry time
   d. scan deleted
   e. time in queue
   f. time in air
   g. total time in area
2. Histogram of time between entries in take-off queue--departures
3. Histogram of actual time between departures
4. Histogram of imposed delay in take-off queue--departures
5. Histogram of time between successive departures from terminal area
6. Histogram of actual flight time in terminal area--departures
7. Histogram of total time in terminal area--departures
8. Histogram of time between entries in enroute queue--arrivals
9. Histogram of actual time between arrivals
10. Histogram of imposed delay in enroute queue--arrivals
11. Histogram of time between successive touchdowns--arrivals
12. Histogram of actual flight time in terminal area--arrivals
13. Histogram of total time in terminal area--arrivals
14. Histogram of range to closest aircraft for the five closest aircraft.

Note: with all histograms except Range to Closest Aircraft, the sample mean, variance, and standard deviation are also calculated.

C. Comparison of Model-Generated Data with Actual Radar Tracking Data

Model validation by comparison with real-world data has been an important consideration from the beginning of the program. Ultimately, the simulation model will be used in what will essentially be experimental investigations. The main aim of these investigations will be to draw inferences about the behavior of some particular ATC environment based on the observed behavior of the model. Consequently, proper usage of the model will require use of all the rules of physical experimentation, plus others arising out of the special nature of computer simulation. Further, in order to draw inferences about the real environment, it is necessary to assure that there is a measure of agreement between the behavior of the model and that of the real system. However, it is important to recognize at the outset that as in most simulation applications, our model will be constantly transformed into a variety of different "versions," as it gets used in the analysis of different procedures, regulations, traffic patterns, instrumentation, etc. Many of these versions will model environments which are not in existence, so that true validation in the above sense is not possible. Hence, when we speak of validation we speak of demonstrating that for at least one version of the model, the behavior of the model is similar to the behavior of a real system. As Conway, Johnson and Maxwell point out [Ref. 3],

> the establishment of validity for one variation and one set of conditions does little to establish the virtue of other variations of the model. So this test, which is widely and quite reasonably applied, is essentially a null test: a model which failed to pass would be exceedingly suspect, but no strong statement can be made for a model which passed.

Given this caveat, we now proceed to explore validation procedure.

"Behavior" of the simulation model has to be defined in terms of specific variables of interest generated as a result of simulation. These variables consitute output of the simulation, and all simulation outputs are time-series. Examples are: "operations" per hour, time in terminal airspace, time in hold, range to closest aircraft, time-to-closest approach, etc. Corresponding variables in the real environment are also time-series. The essential problem of validation, then, is to show that the simulation time-series are, to a degree of approximation, the same as the corresponding real environment time-series with regard to statistical characteristics.

Statistical procedures involving estimation and confidence levels require the assumption that the data come from stationary distributions*, and this has a major impact on validation procedure. Consider for a moment the simulated time-span. Any problem-solving run of the simulator will involve its operation over a period of simulated time T, which we call the simulated time-span. The simulated time-span T consists of two portions:

1. There is a run-in time during which the output variables have not reached "steady state" and are still reflecting the transients of the start-up conditions. The length of this time can vary, depending on the extent of artificiality in the start-up conditions. If, as usual, the simulator is started "empty" (no planes in the air or on the ground, controllers idle, etc.), run-in time would be longer than if it were started with a load resembling steady-state operations. Statistical considerations require that output data from the run-in phase be ignored for statistical inferences. Thus, since start-up conditions cannot with confidence be conjured up to be in steady state, there should always be some run-in time during which output variables are merely monitored for stability.
2. After run-in is completed, a measurement period begins. This period corresponds to sample size in more conventional experiments, and statistical considerations require that this period be long enough (i.e., there be enough "observations" taken) for means and other parameters to be estimated with a high level of confidence.

The general concept of a run-in period and a measurement period is mentioned in the context of a problem-solving run, but it also applies for validation runs.

In previous studies, analyses had been conducted with actual radar tracking data obtained from the ARTS system at the Atlanta terminal. These data were taken by FAA (NAFEC) personnel over a time span of five days during the summer of 1967. The radar data consisted of x, y, z, position and velocity data on every aircraft under track at Atlanta over twelve one-hour peak traffic periods. The data were taken at four-second intervals throughout each of the (approximately) one-hour sample periods. A more

*Some of the more common procedures require, in addition, that the data come from normal (Gaussian) distributions. However, "distribution-free" methods are also available which do not have this additional requirement. In any case, even the simpler procedures are not much affected by moderate departure from normality.

detailed description of this data base is given in Ref. 4.

To provide a comparison of the radar data with the model-generated data, the model was configured to represent the route structure in use at the time the radar data were taken (i.e., Atlanta 1 Route Structure), and a traffic sample was generated using as input parameters the same average operations per hour, average route loadings, and approximate aircraft type distribution as existed during one of the most dense hours of the available sample periods. Simulation runs were then made to generate a one-hour data sample in the same format as the Atlanta radar data.

Figure 19 shows a comparison of a simulated (model-generated) track and an actual radar track of a B727. The initial starting position of the aircraft is slightly different; however, both aircraft pass over the OCR VOR site. As may be seen in this particular example, the tracks compare favorably in appearance and in that the total flying time of both paths is approximately the same. Simulated path deviations appear realistic, and in most cases investigated, similar good comparison of individual tracks was noted.

To investigate the relationships among the individual aircraft tracks to determine if realistic relative separations were being maintained and if the overall control scheme was operating realistically, statistics were generated to determine the average percentage of flying time that an aircraft found another aircraft within a range interval ΔR. This statistic provides an estimate of the probability that a randomly selected aircraft from the aircraft population will have another aircraft within a range interval ΔR. Calculations required to develop an unbiased estimate of this statistic are given in Ref. 4.

Figure 20 indicates the statistical results of comparing the model-generated and real-world data on this basis. As may be seen, the results compare favorably in all range bins with the possible exception of the 2 to 3 mile range where the model data indicate a slightly higher percentage of flying time took place than occurred in reality.

A parameter that has been proposed as a warning criterion in collision avoidance systems by several investigators and is recommended as a warning criterion in the ATA Specifications for a Collision Avoidance System is a useful parameter to use to estimate the overall safety level of the system. This parameter is the τ parameter, or the approximate time to closest approach of a pair of aircraft. Thus, statistics have been generated on

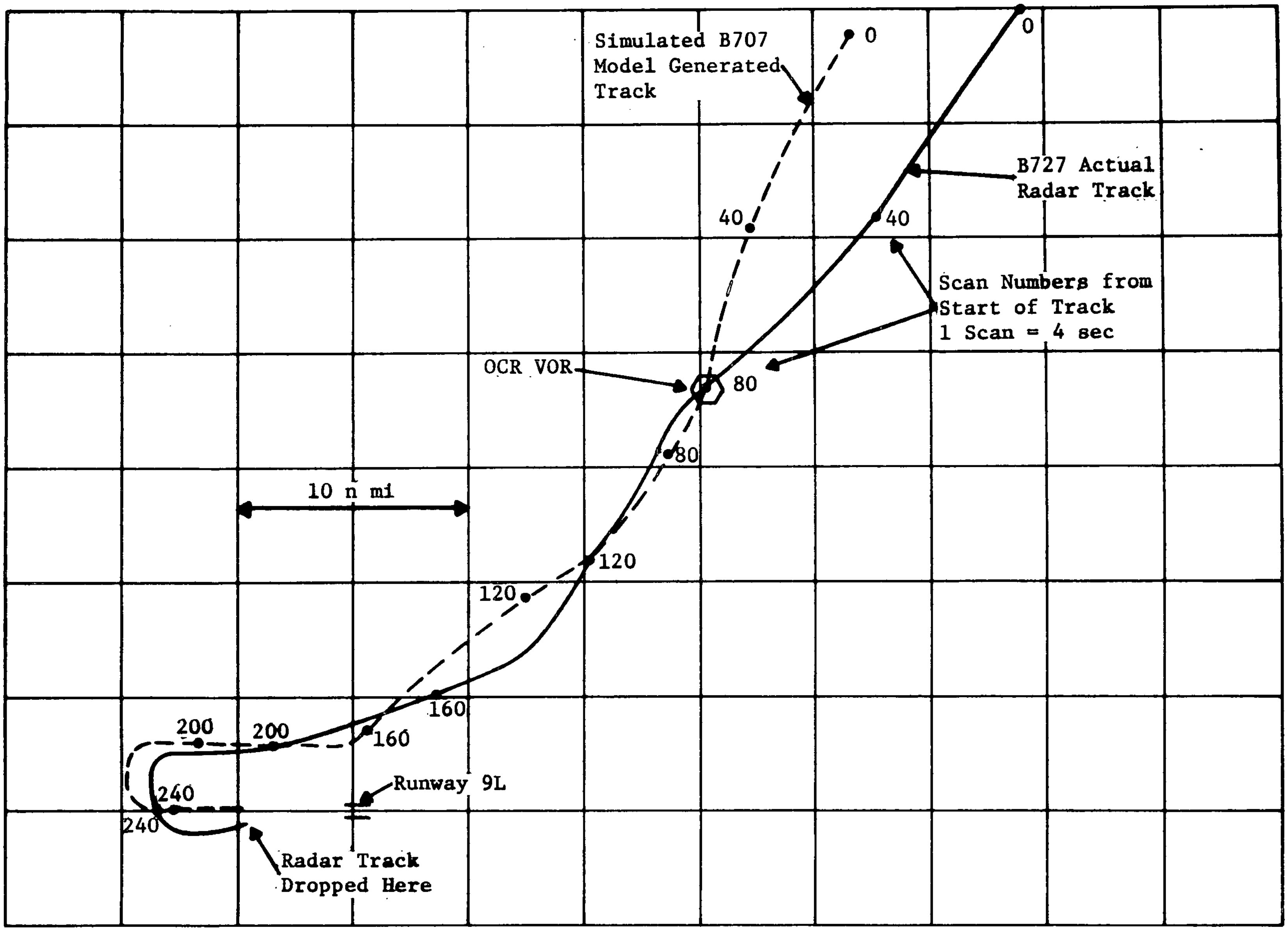
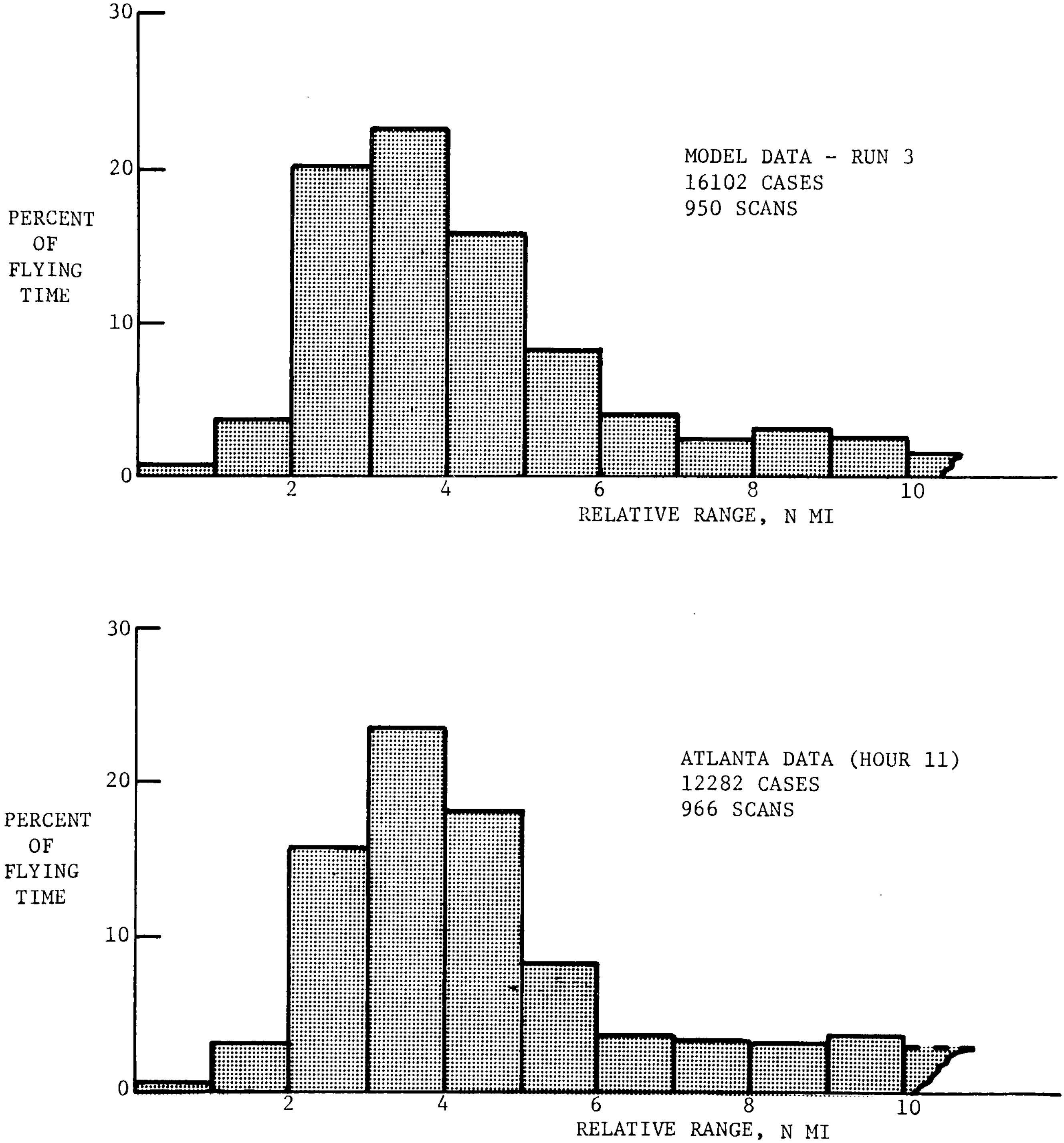


Fig. 19. Comparison of a model-generated aircraft track with an actual radar track of a B727 from the Atlanta radar data.



DISTRIBUTION OF RANGE TO CLOSEST AIRCRAFT

Fig. 20. Histograms showing the percentage of flying time that a randomly selected aircraft found another aircraft within the range bins specified on the horizontal axis for model-generated data and Atlanta radar data.

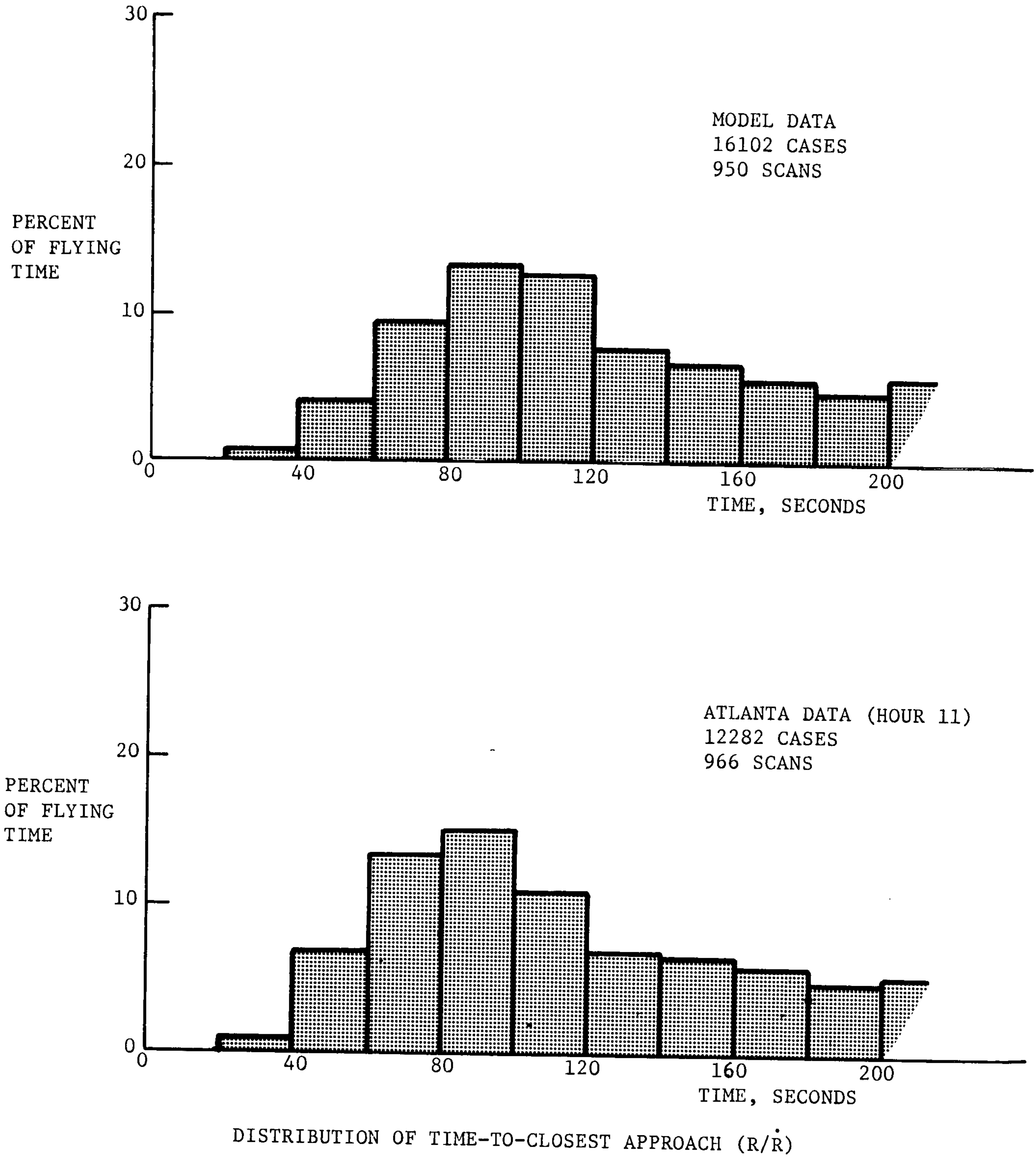


Fig. 21. Histograms showing the percentage of flying time that a randomly selected aircraft found another aircraft within a time-to-closest approach specified on the horizontal axis for model-generated data and Atlanta radar data.

the τ parameter which indicate the probability that a randomly selected aircraft will find another aircraft within a time-to-closest approach of τ seconds. A histogram showing the distribution of the τ parameter for model data and Atlanta radar data is shown in Fig. 21. The comparison indicates that the distribution of times-to-closest approach is similar in shape with variations in individual histogram bins on the order of a few percent.

The comparison of the model-generated statistics with real-world statistics indicates, to a high degree of approximation, that the model as configured does in fact provide a good representation of the real-life environment. It would be highly desirable to have additional real-world data for further comparisons; however, to date attempts to obtain additional real-world data have been unsuccessful.

D. Test Runs to Determine Sensitivity to Random Aircraft Heading Errors

To illustrate the type of sensitivity studies to be conducted with the model, two test runs were made under identical conditions except that in one case, random heading errors of approximately 3° rms were introduced for all aircraft. A traffic sample representing approximately 70 operations per hour was used and the model was configured to represent Atlanta using a single runway.

Table 7 illustrates the effect of the random heading errors on the aircraft flow parameters. The most sensitive parameter under the conditions simulated was the actual rate of departures achieved. Note from the table that while 20 departures per hour were scheduled, a rate of only 14.3 per hour was achieved in the no-error case, whereas 16.7 departures were achieved in the case with heading errors. The above results are somewhat surprising, however, a study of the remaining data in the table indicates that the main effect of the heading errors was to slightly decrease the rate of arrivals, hence permitting more departures to be cleared. The rms deviation of the total flight time for arrivals increased slightly in the case with errors. For the case in which heading errors were introduced, the relative range distributions (i.e., as in Fig. 20) indicated a slightly higher probability of finding an aircraft within a two-mile range.

The above results should not be generalized, since they are based on sample runs of approximately 90 simulated minutes under a particular set of convenient conditions. Rather, the data are presented to demonstrate the type

Table 7. Statistics developed for comparison of sample runs with and without random heading errors. Each run represents approximately 90 simulated minutes at approximately 70 operations/hr.

		Scheduled	Sample Run with 3° rms random heading errors	Sample Run with no heading error
1.	Average time between departures	2.9 min.	3.6 min.	4.2 min.
2.	Average departure rate	20/hr	16.7/hr.	14.3/hr.
3.	Average waiting time for departures	-	4.8 min.	7.6 min.
4.	Average time between arrival touchdowns	1.28 min.	1.33 min.	1.30 min.
5.	Average rate of touchdowns	47/hr	45.1/hr.	46.2/hr.
6.	RMS variation of time between touchdowns	-	.4 min.	.5 min.
7.	Average flying time in the terminal area for arrivals	-	21.4 min.	21.2 min.
8.	RMS variation of total flying time for arrivals	-	3.5 min.	3.0 min.
9.	Average enroute holding time for arrivals	-	26.4 sec.	26.4 sec.
10.	Average percentage of flying time that a randomly selected aircraft had another aircraft within 2 n.mi.	-	4.3%	3.6%
11.	Average percentage of flying time that a randomly selected aircraft had another aircraft within 4 n.mi.	-	41.9%	46.4%

of answers that will be obtained using longer runs that permit the development of more statistically significant data under carefully selected conditions.

Preceding page blank

VI. CONCLUSIONS

A terminal area simulation model has been defined that contains the major features of the air traffic control situation in the terminal area. Algorithms have been developed which permit the simultaneous generation of the tracks of many different aircraft from entry into the problem area to touchdown on the runway (for arrivals) and from the engine run-up point on the runway to exit from the problem area (for departures). The paths are defined by controller-generated radar vectors and by navigation aids that are currently used by pilots in the terminal area.

In addition, techniques have been developed for conflict detection, conflict resolution, sequencing, and for the generation of the associated pilot-controller messages. These messages incorporate realistic delays and are constrained by the channel loadings of the controller-pilot communication links. Associated display techniques have been developed which permit real-time observation of the traffic flow with a simulated radar CRT display.

The initial model has been programmed for the CDC-6600 computer and run in both real-time and fast-time on the computer complex at Langley Research Center. The initial test runs indicate that the model and associated program are within the capabilities of the computer facilities available. The test runs also indicate good correlation with actual radar traffic data obtained by the FAA at the Atlanta airport.

As the initial model development progressed, certain areas of improvement were noted but were not implemented because of time limitations. These improvements are now being made, and include the incorporation of more detail in certain areas of the program (e.g., additional flight modes and more flexible route design). Additional logic is being developed to handle unusual situations such as missed approaches and emergency situations.

The model as finally developed is intended to be general and flexible, such that changes in instrumentation, terminal area configuration, and aircraft performance characteristics can be incorporated and the effects of these changes on the overall system evaluated by analysis of the output data.

The principal uses of the terminal area model and associated simulation facility will be to determine the effects of new aircraft types, improved

avionics and ground instrumentation, and automated sequencing, spacing and flow control on congestion, operational capacity and safety. The facility will also be used to provide a realistic multiple-aircraft environment for the evaluation of pilot displays, terminal optimized vehicle configurations, collision-hazard warning system techniques, and advanced landing guidance systems.

VII. APPENDICES

Preceding page blank

APPENDIX A

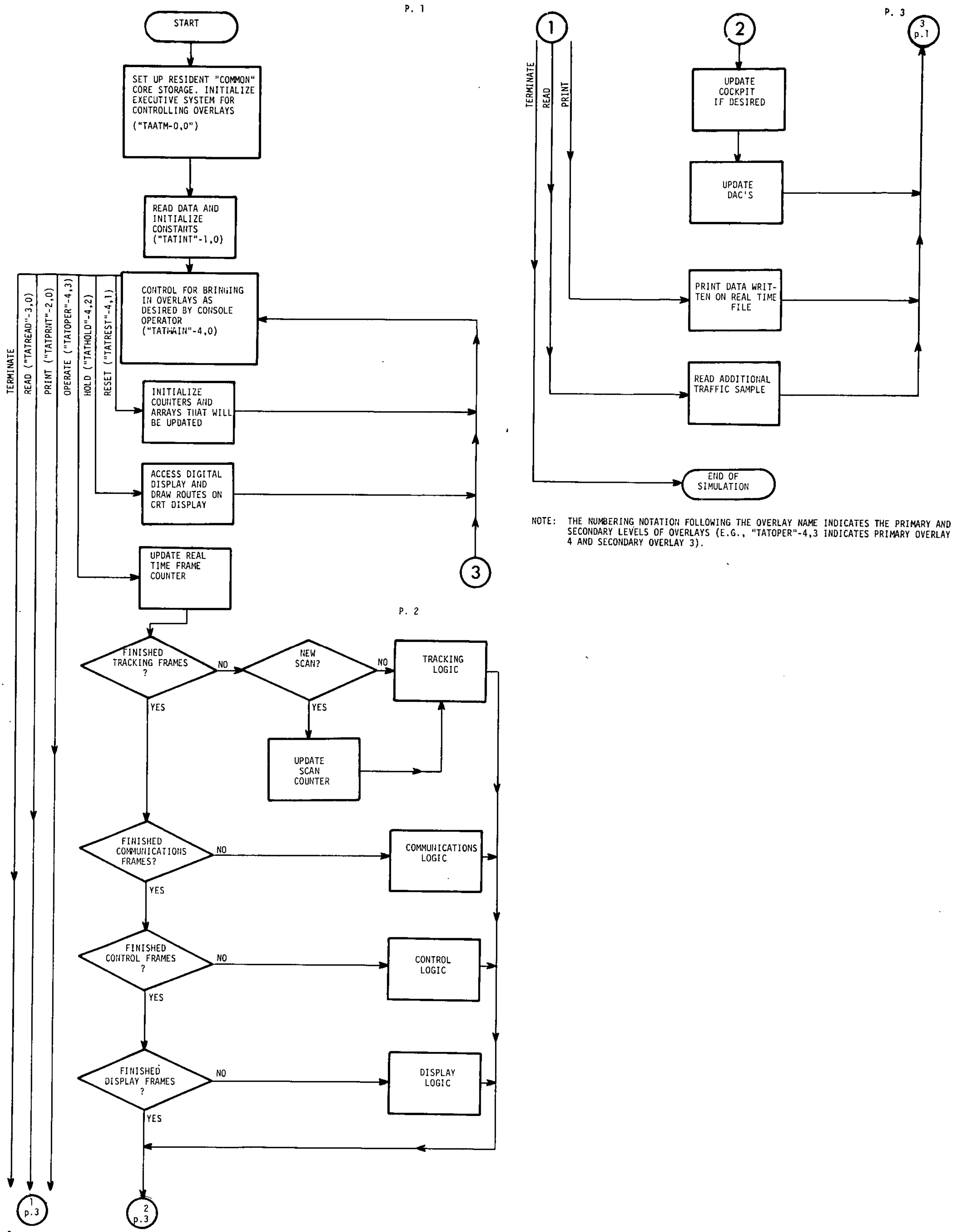
Terminal Area Simulation Model Logic Flow Charts

This appendix contains flow charts in which the logic of the model is described with sentences or phrases in English for the decision and process boxes. The detailed coding and specific algorithms used are included in Appendix H.

Statement numbers are included outside particular flow charting symbols for easy reference back to the corresponding statement numbers in the programs themselves. Relative statement numbers, noted by a + or -, indicate the number of FORTRAN statements (excluding comments) below or above the given statement number; e.g., 3675+8 points to the statement which is eight FORTRAN statements below the statement numbered 3675.

The flow charts are grouped in four categories. Figure A-1 is the diagram of the overall organization of the model, and Figs. A-2, A-3, and A-4 contain flow charts for the Simulation Model, the Traffic Generation Program, and the Analysis Program, respectively. The programs and routines included in this appendix are as follows.

PRECEDING PAGE BLANK NOT FILMED



NOTE: THE NUMBERING NOTATION FOLLOWING THE OVERLAY NAME INDICATES THE PRIMARY AND SECONDARY LEVELS OF OVERLAYS (E.G., "TATOPER"-4,3 INDICATES PRIMARY OVERLAY 4 AND SECONDARY OVERLAY 3).

Fig. A-1. Flow chart for the overall organization of the simulation model.

Fig. A-2. Logical flow charts for the Simulation Model.

P. 1

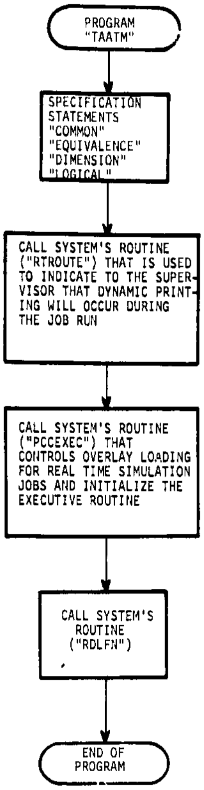


Fig. A-2.1. TAATM - main overlay.

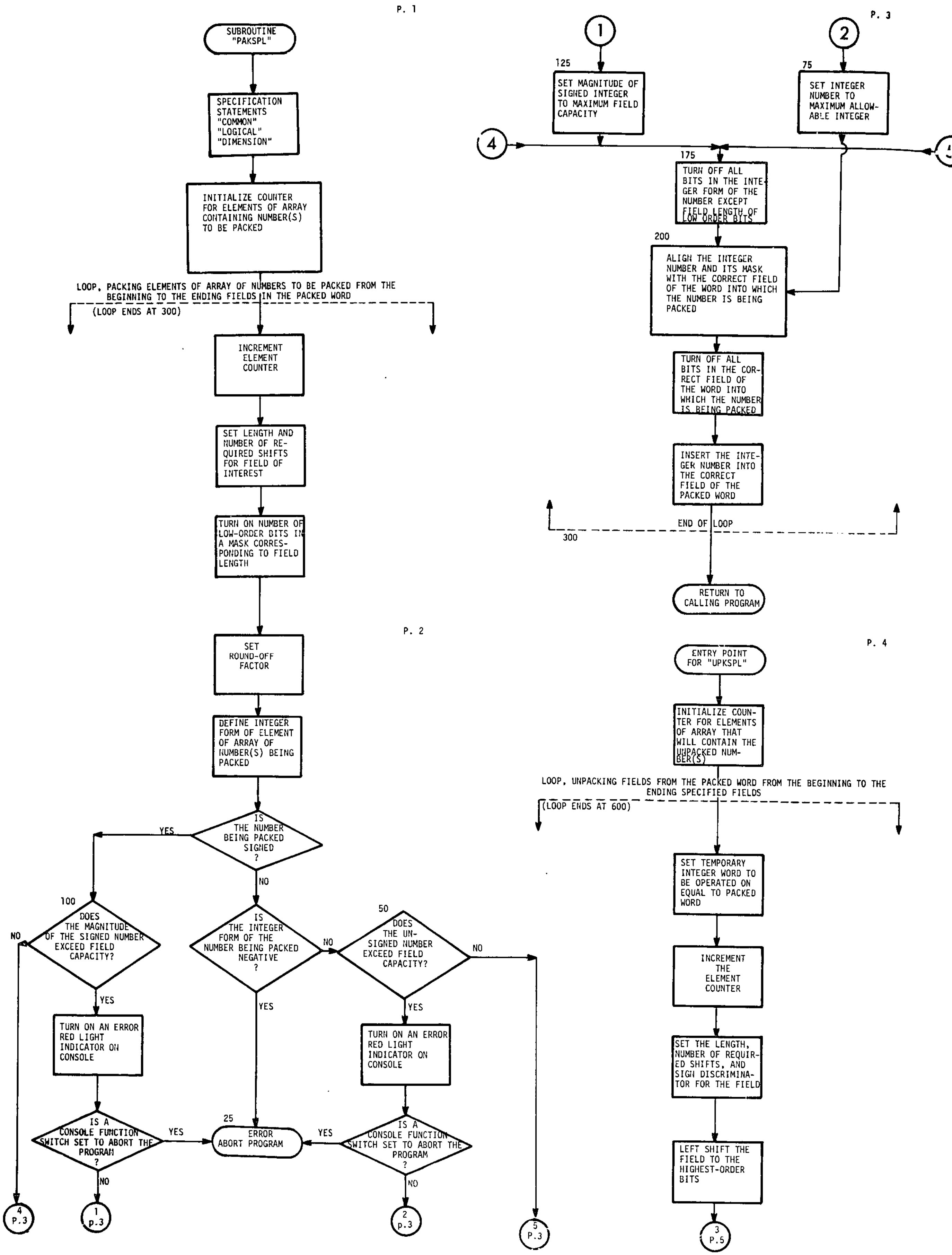


Fig. A-2.2. PAKSPL - pack/unpack special.

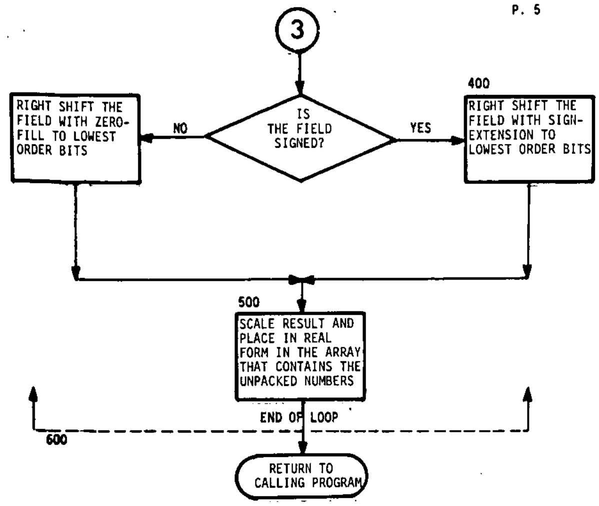


Fig. A-2.2. Concluded.

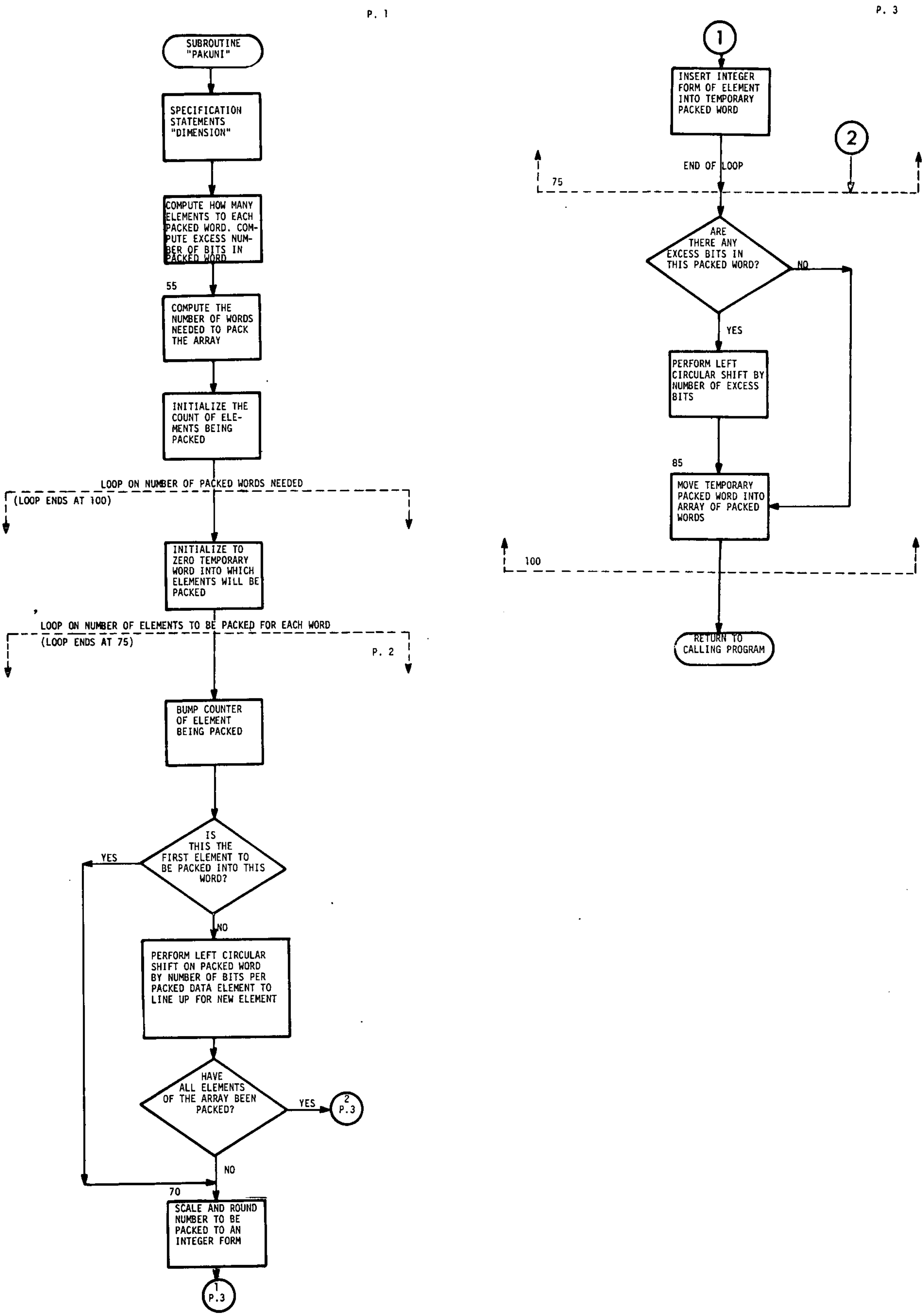


Fig. A-2.3. PAKUNI - pack uniform.

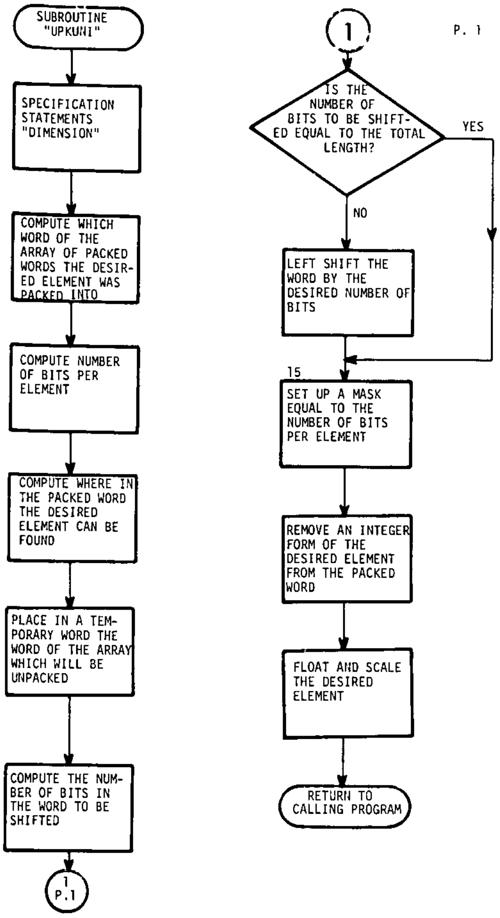


Fig. A-2.4. UPKUNI - unpack uniform.

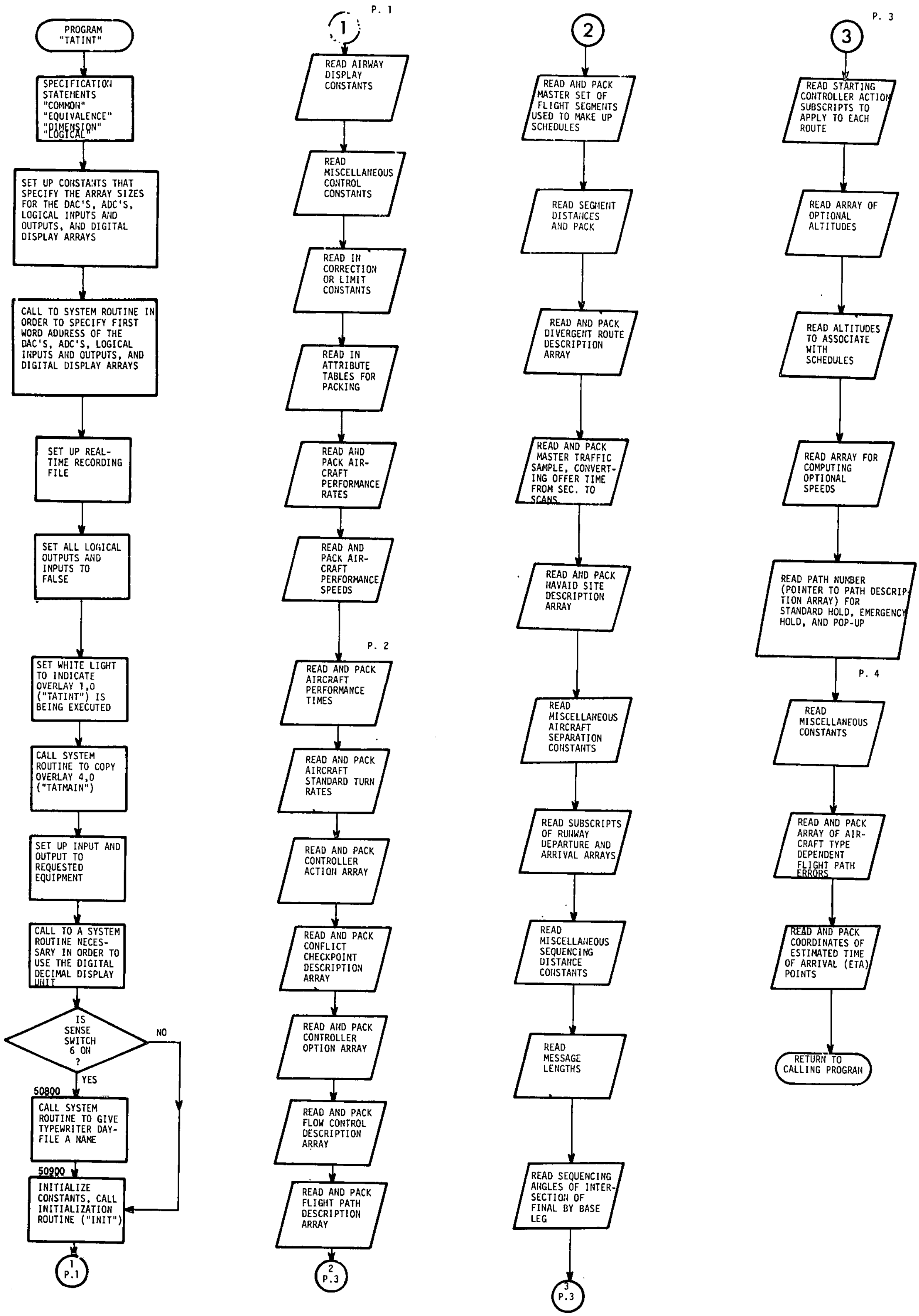


Fig. A-2.5. TATINT - initial read.

P. 1

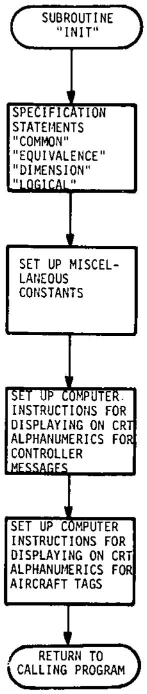


Fig. A-2.6. INIT - initialize.

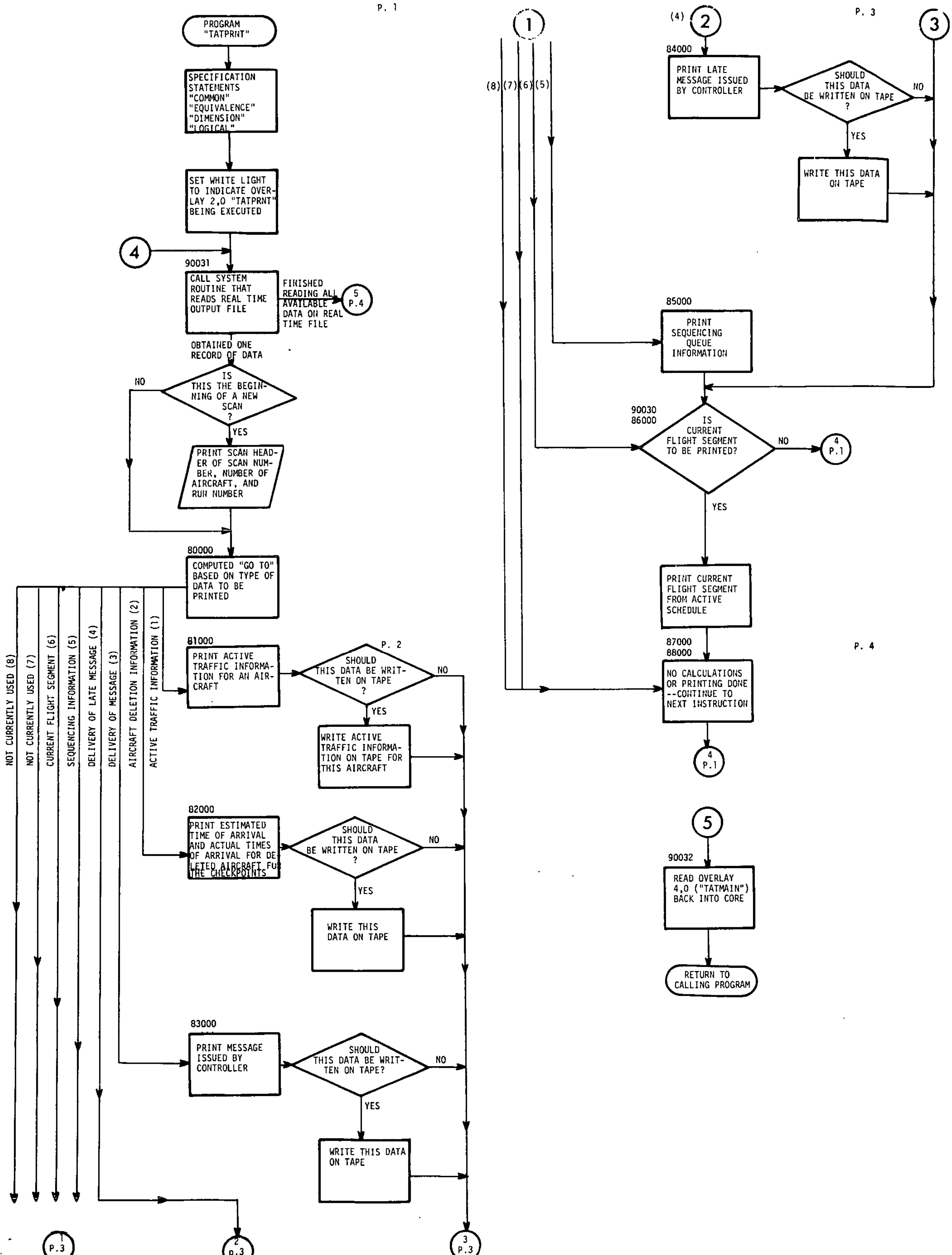


Fig. A-2.7. TATPRNT - print (and tape write).

P. 1

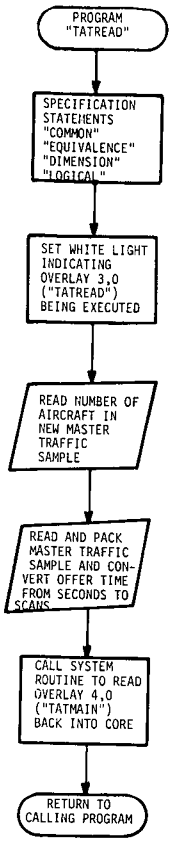


Fig. A-2.8. TATREAD - read additional traffic.

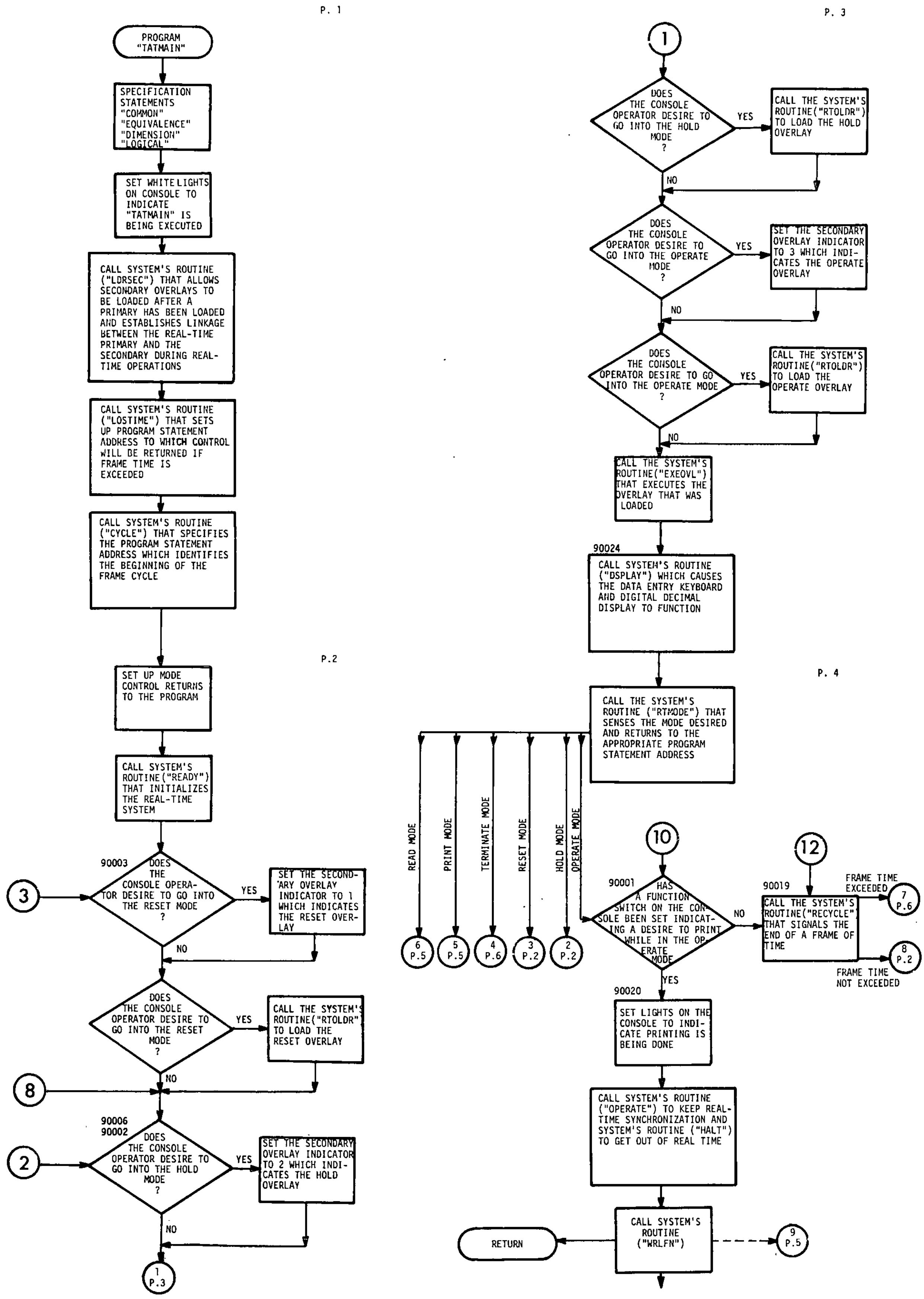


Fig. A-2.9. TATMAIN - main program of overlay 4.

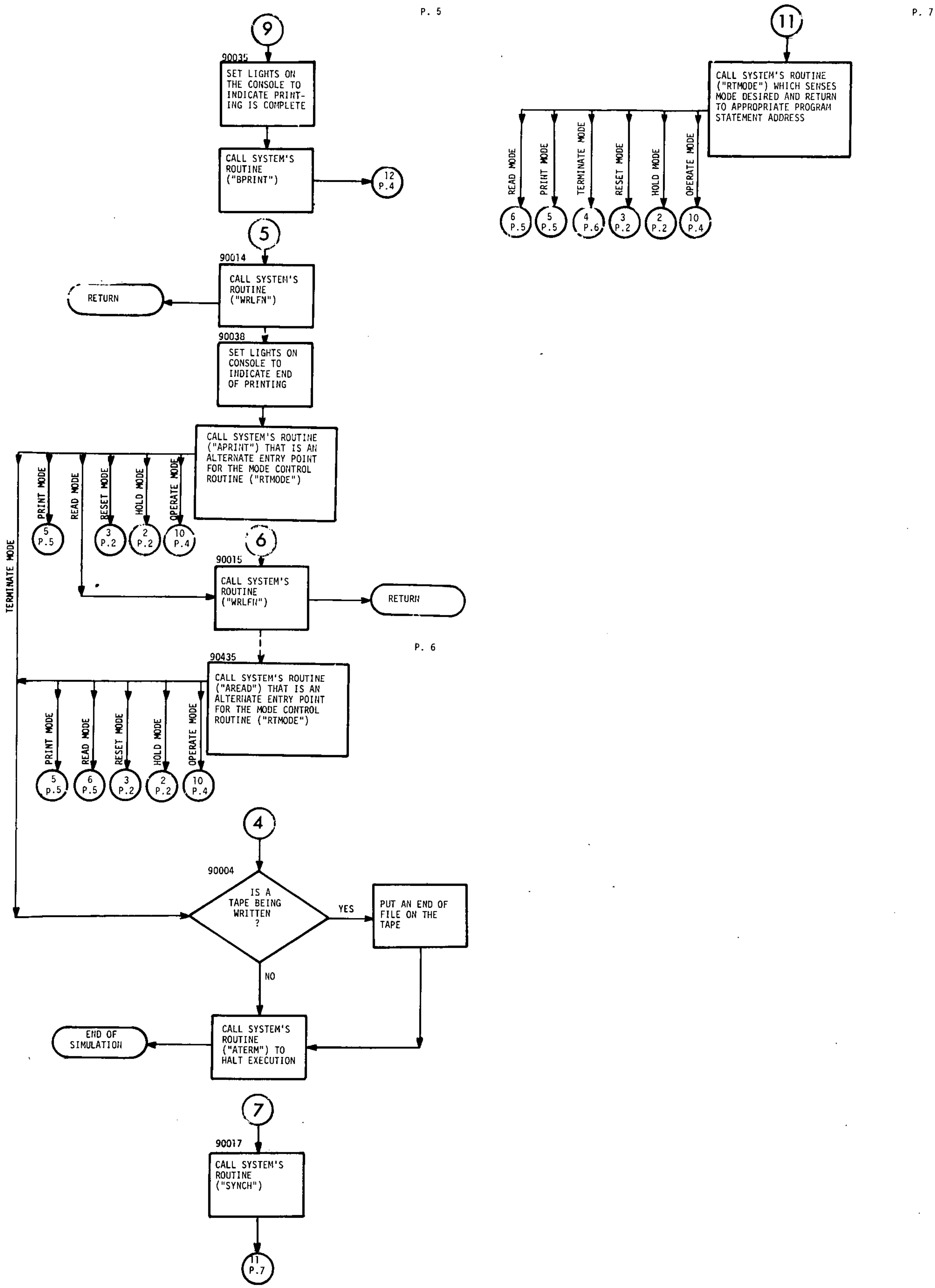


Fig. A-2.9. Concluded.

P. 1

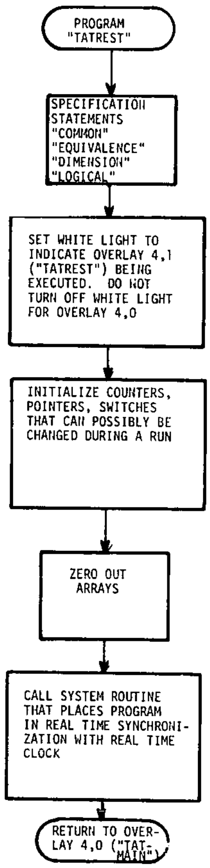


Fig. A-2.10. TATREST - reset.

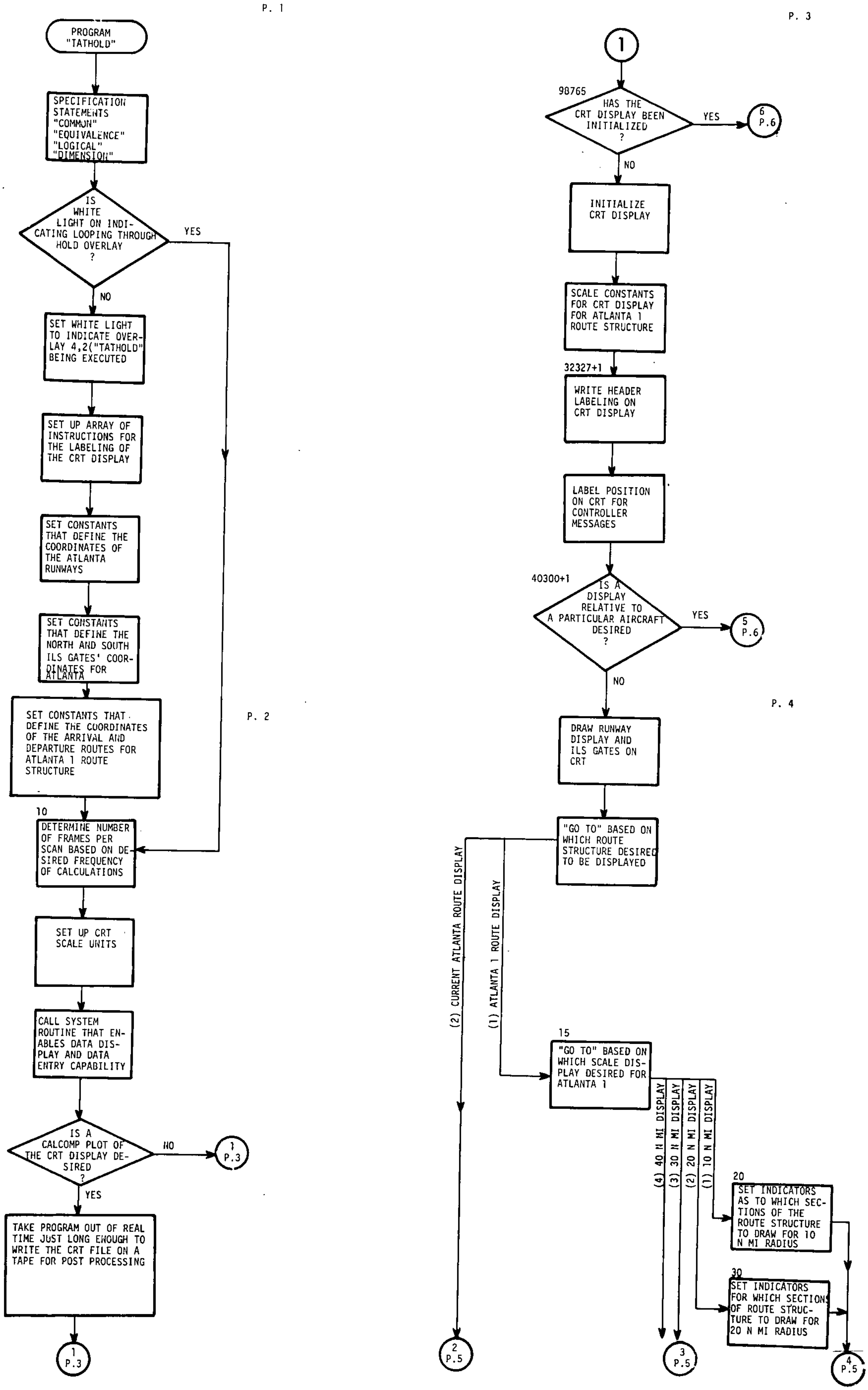


Fig. A-2.11. TATHOLD - hold.

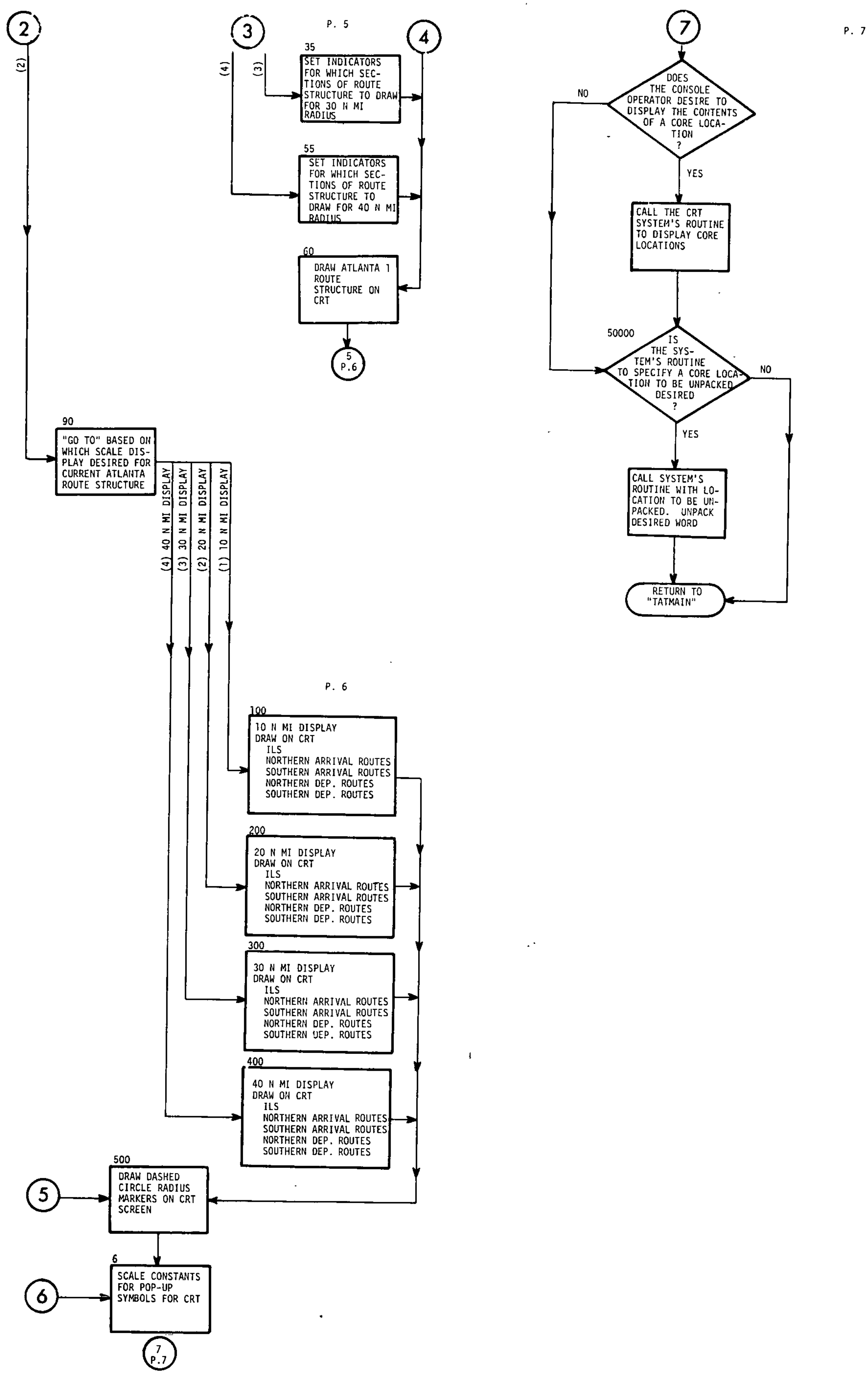
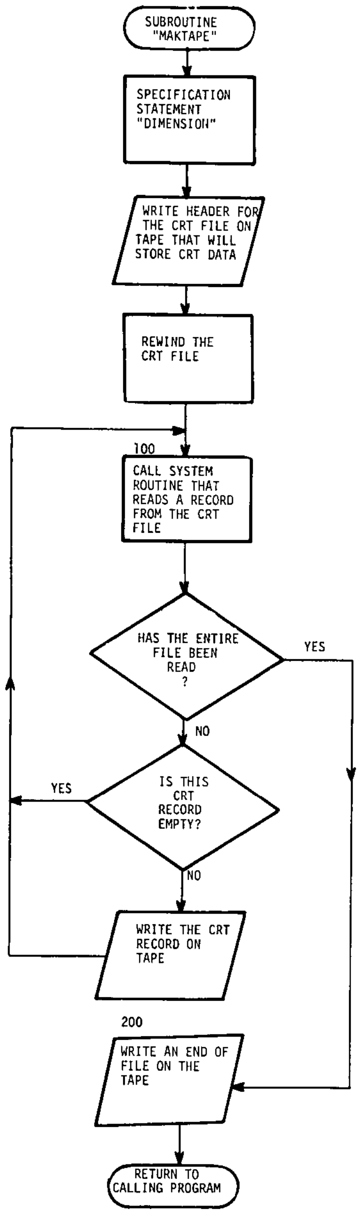


Fig. A-2.11. Concluded.

P. 1



P. 2

Fig. A-2.12. MAKTAPE - make Calcomp plotter tape.

P. 1

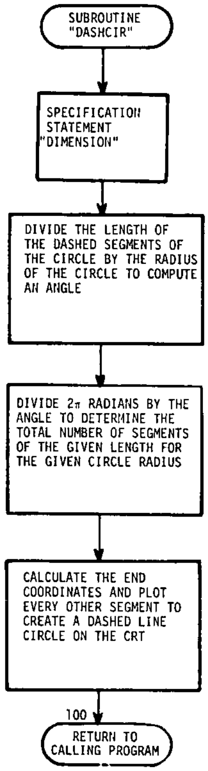


Fig. A-2.13. DASHCIR - dashed circle.

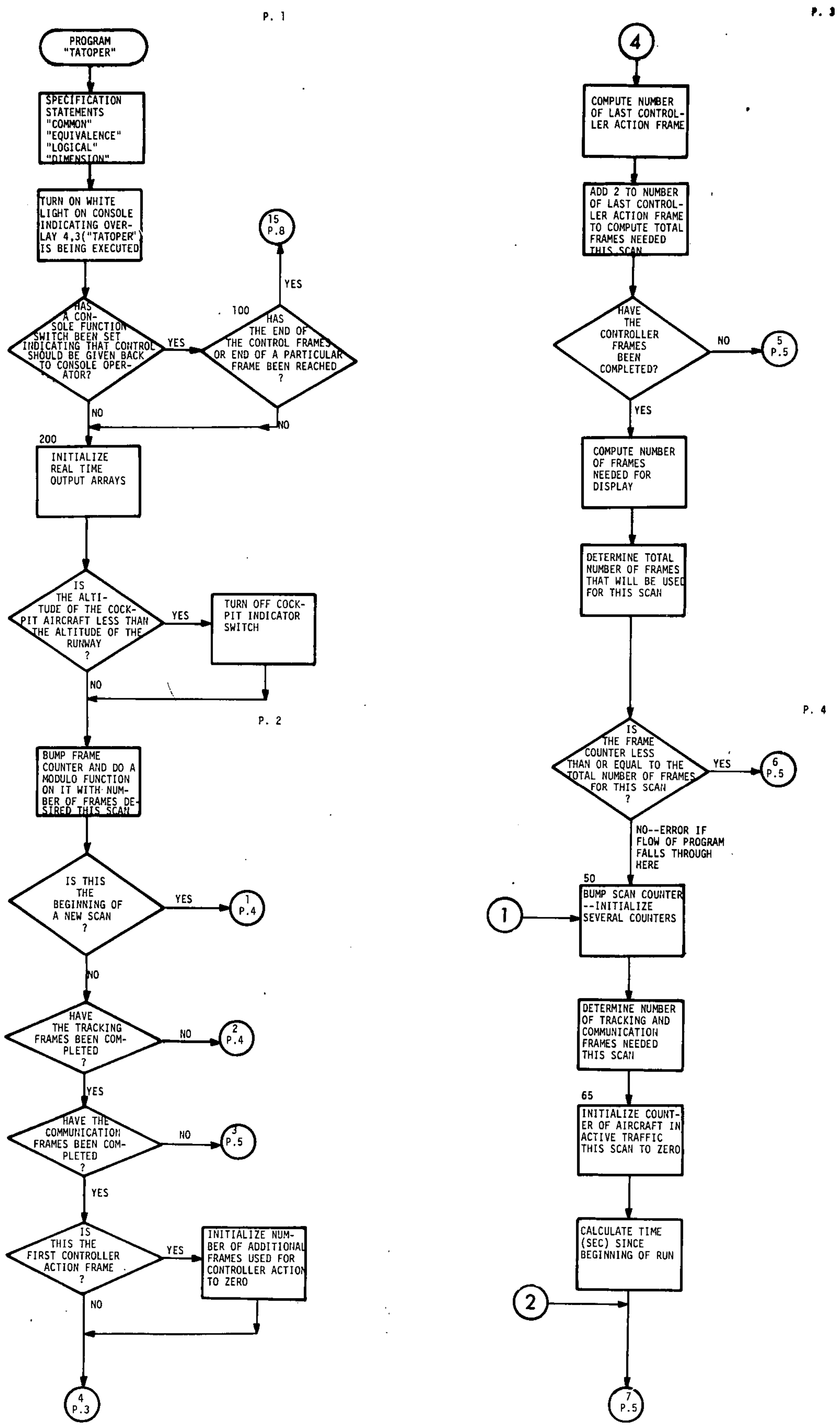


Fig. A-2.14. TATOPER - operate.

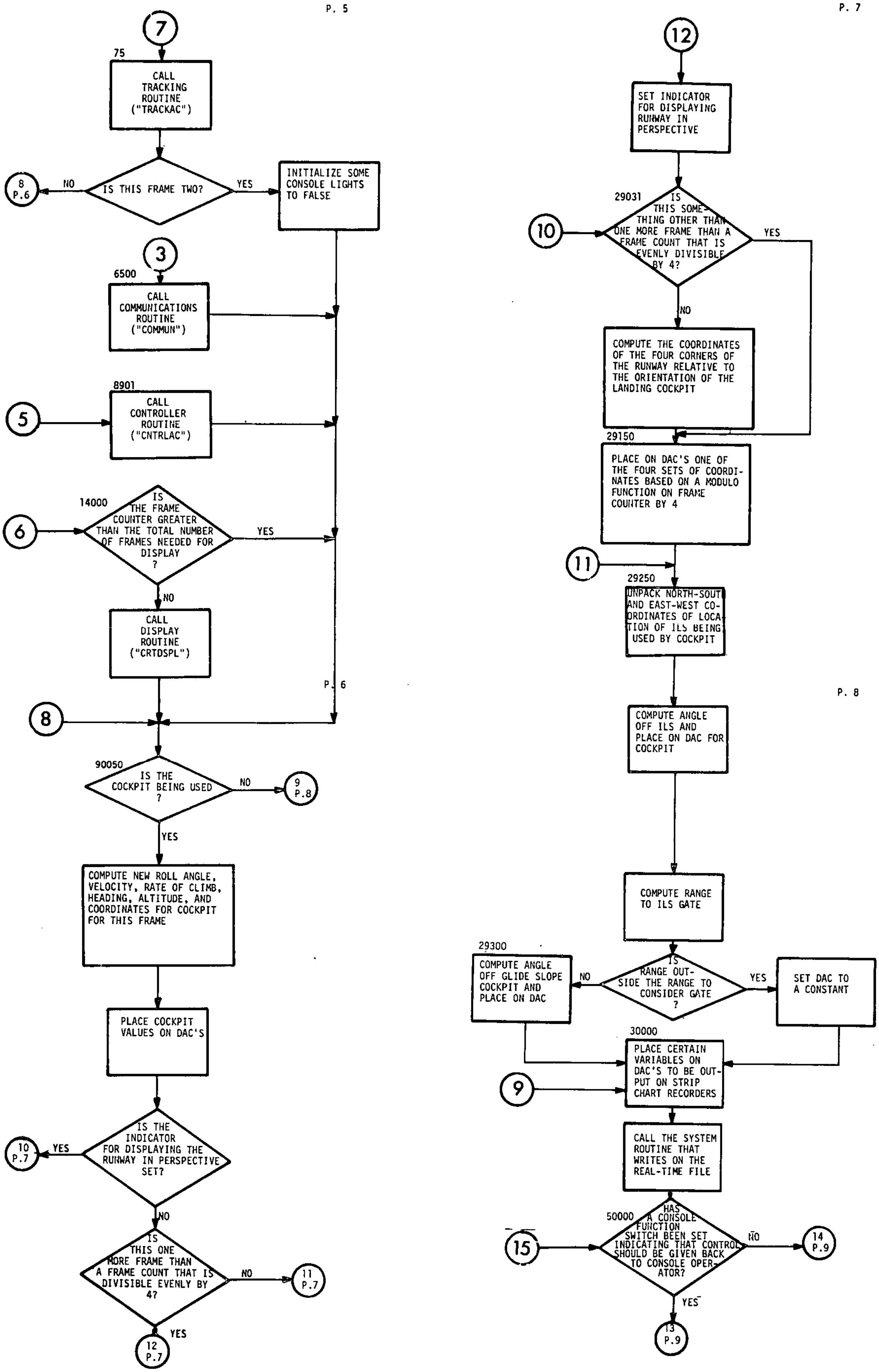


Fig. A-2.14. Continued.

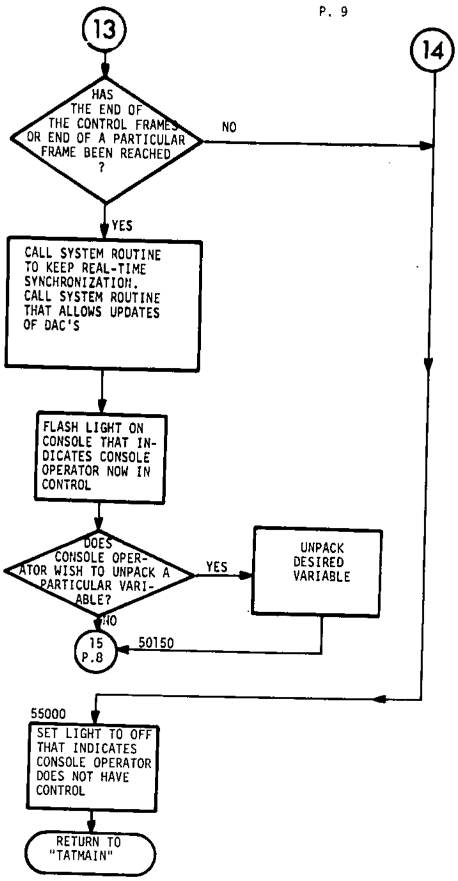


Fig. A-2.14. Concluded.

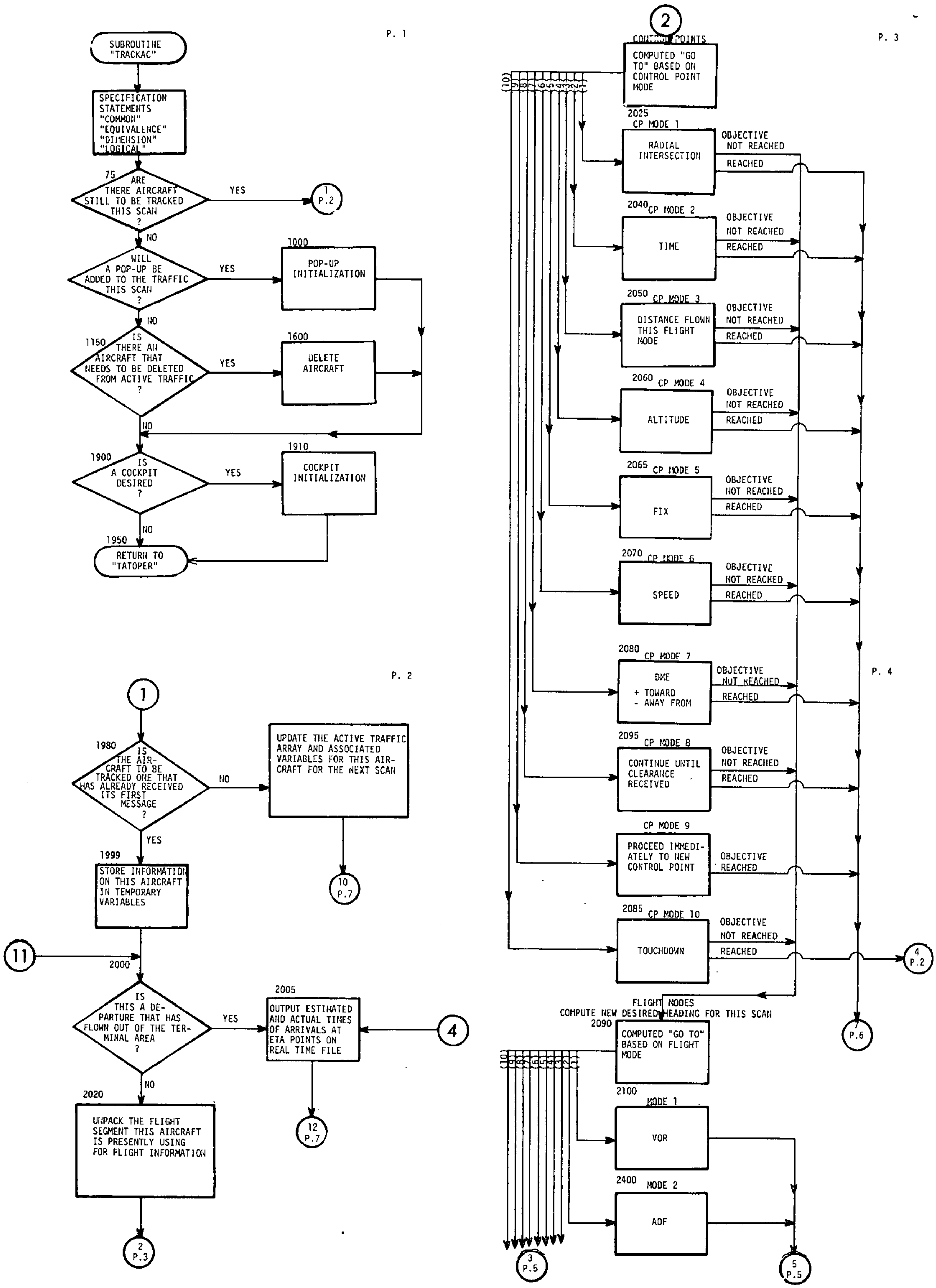


Fig. A-2.15. TRACKAC - track aircraft.

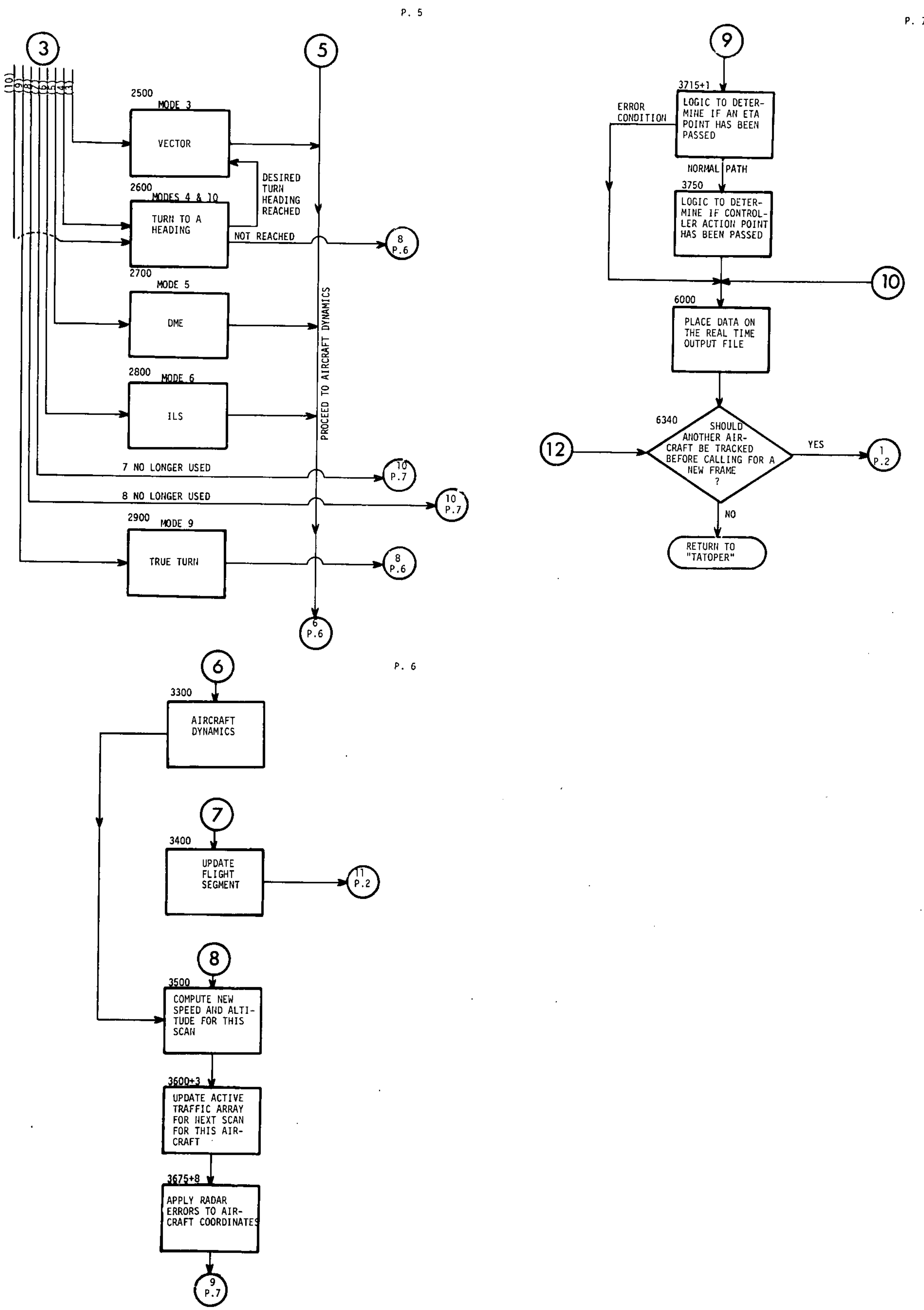


Fig. A-2.15. Concluded.

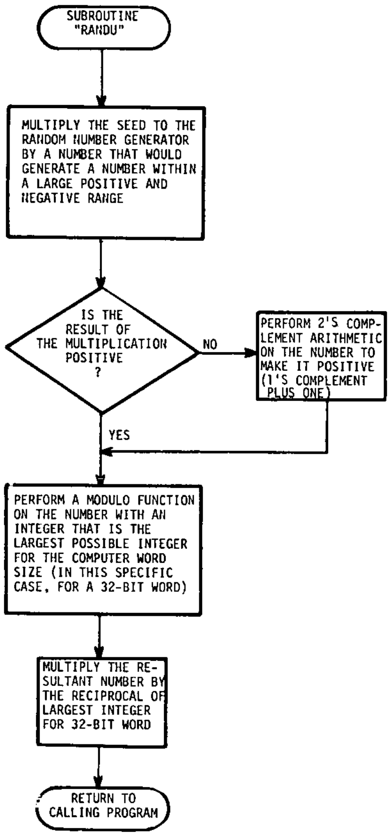


Fig. A-2.16. RANDU - random number generator.

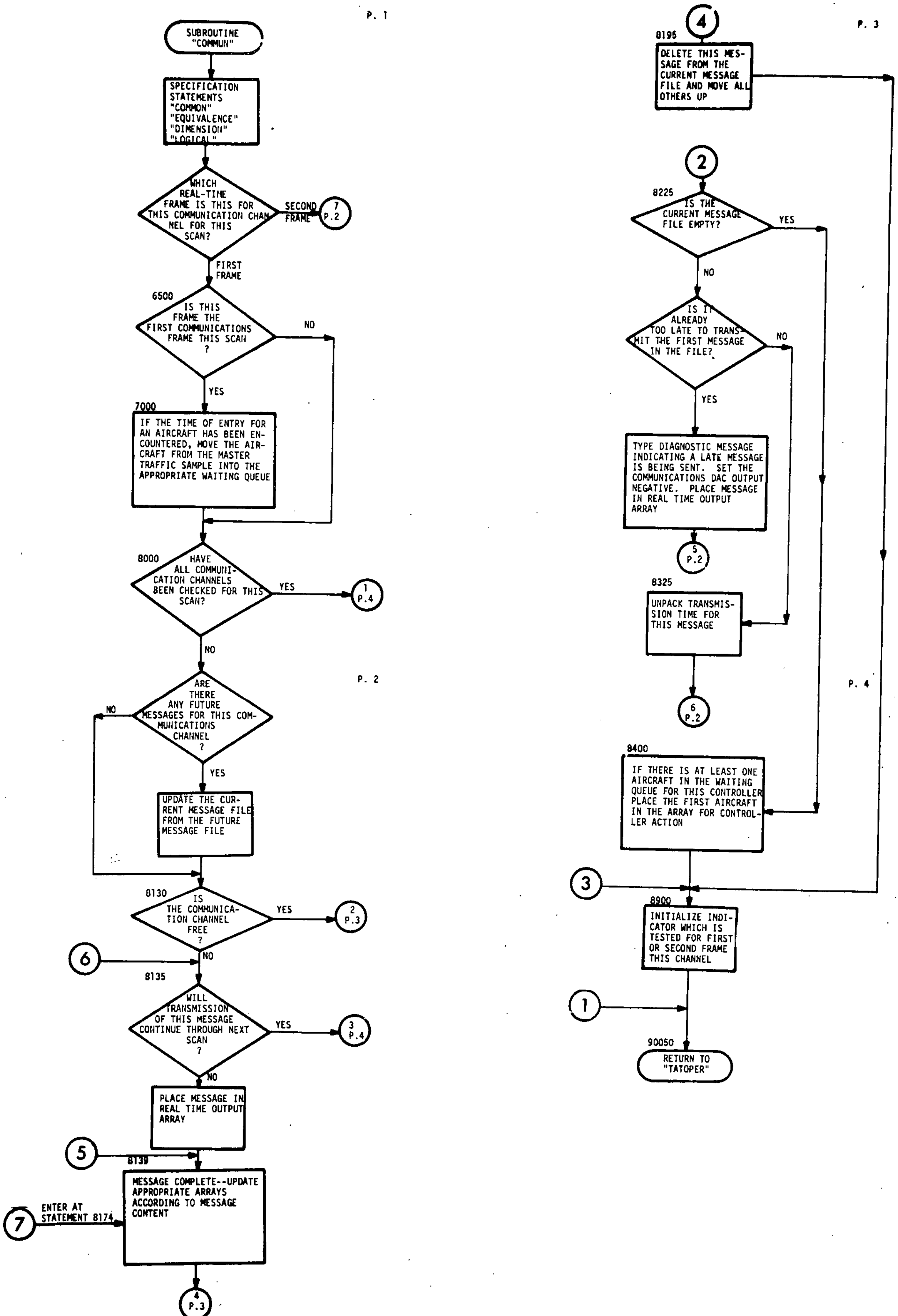


Fig. A-2.17. COMMUN - communications.

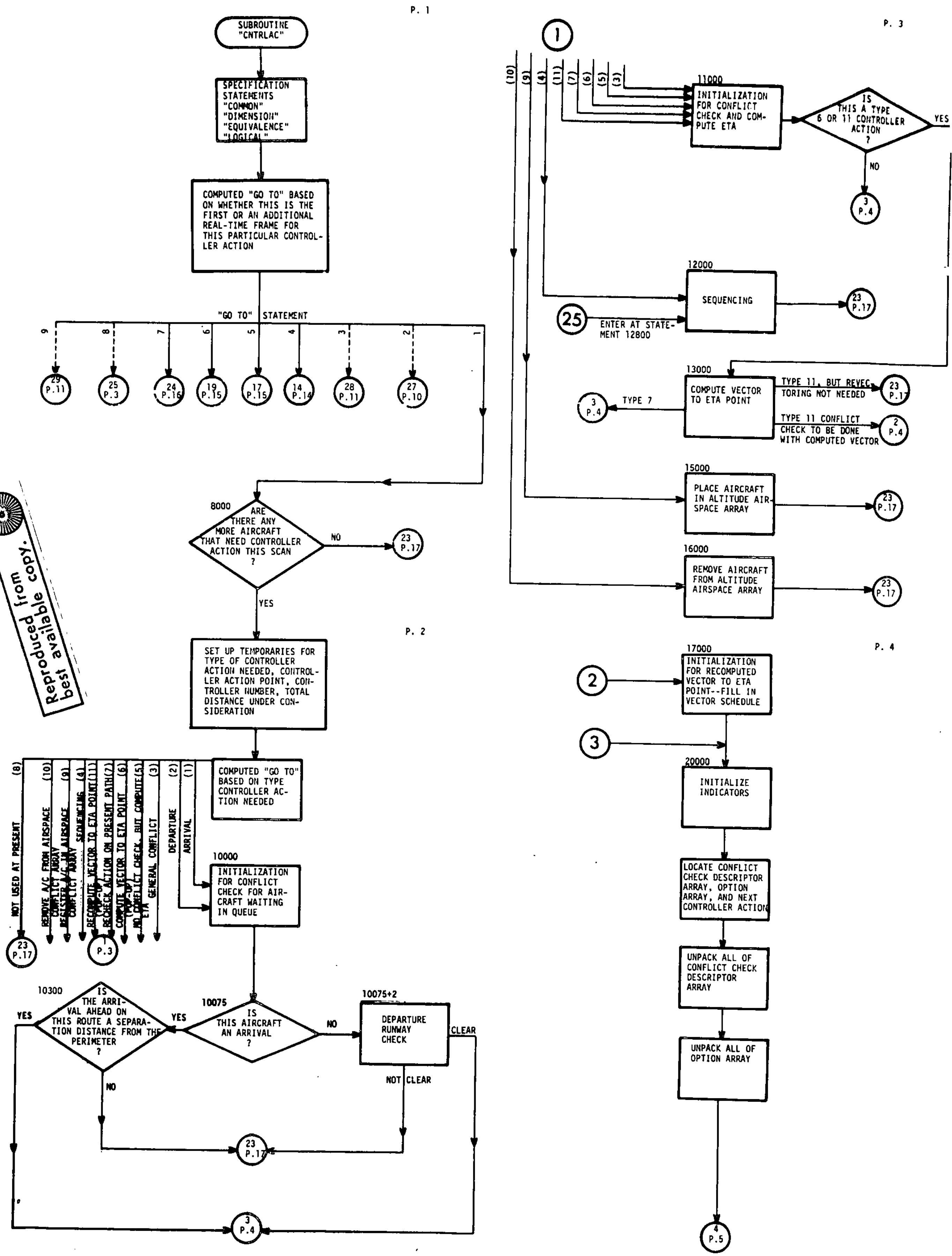


Fig. A-2.18. CNTRLAC - control aircraft.

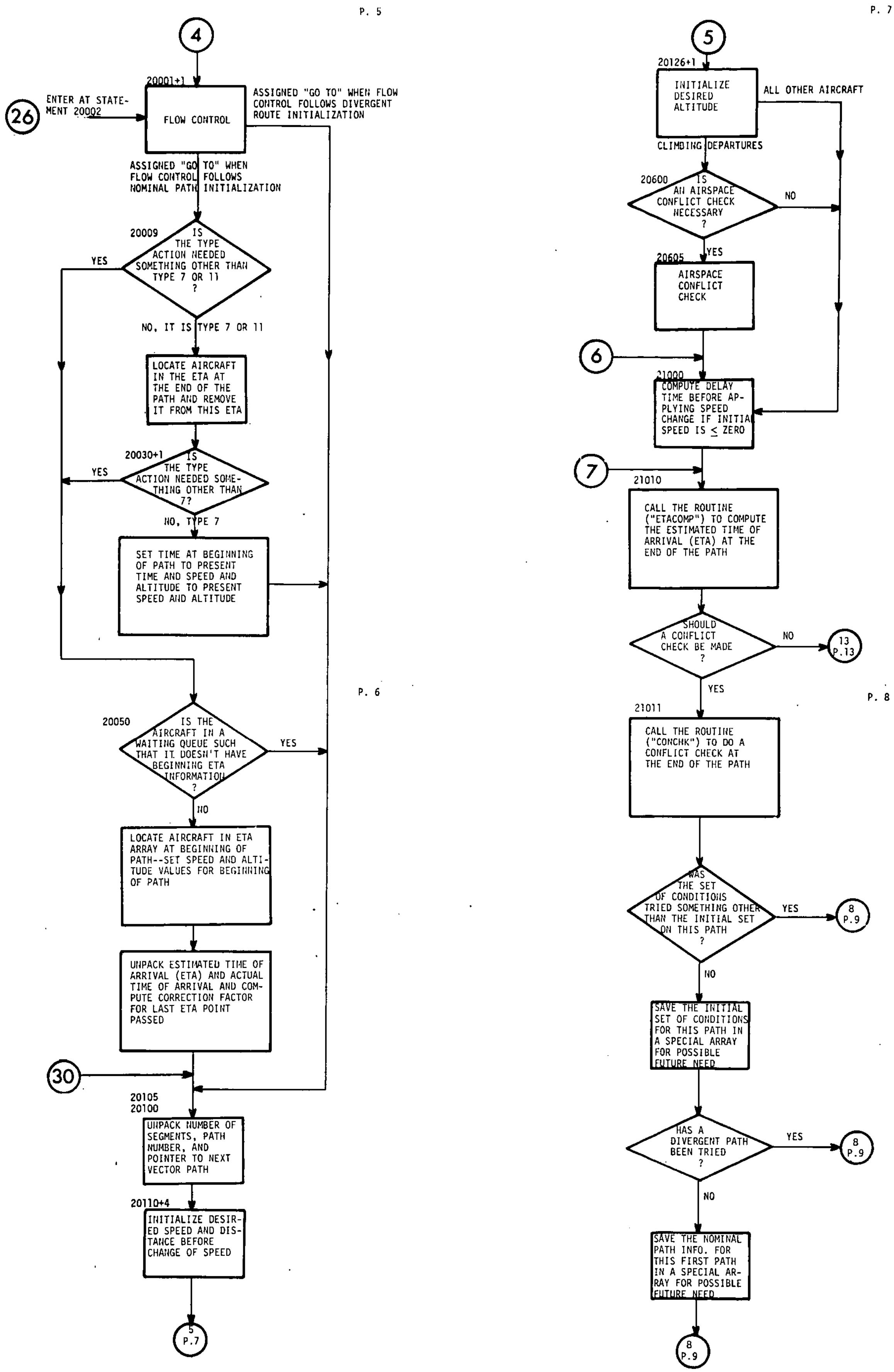


Fig. A-2.18. Continued.

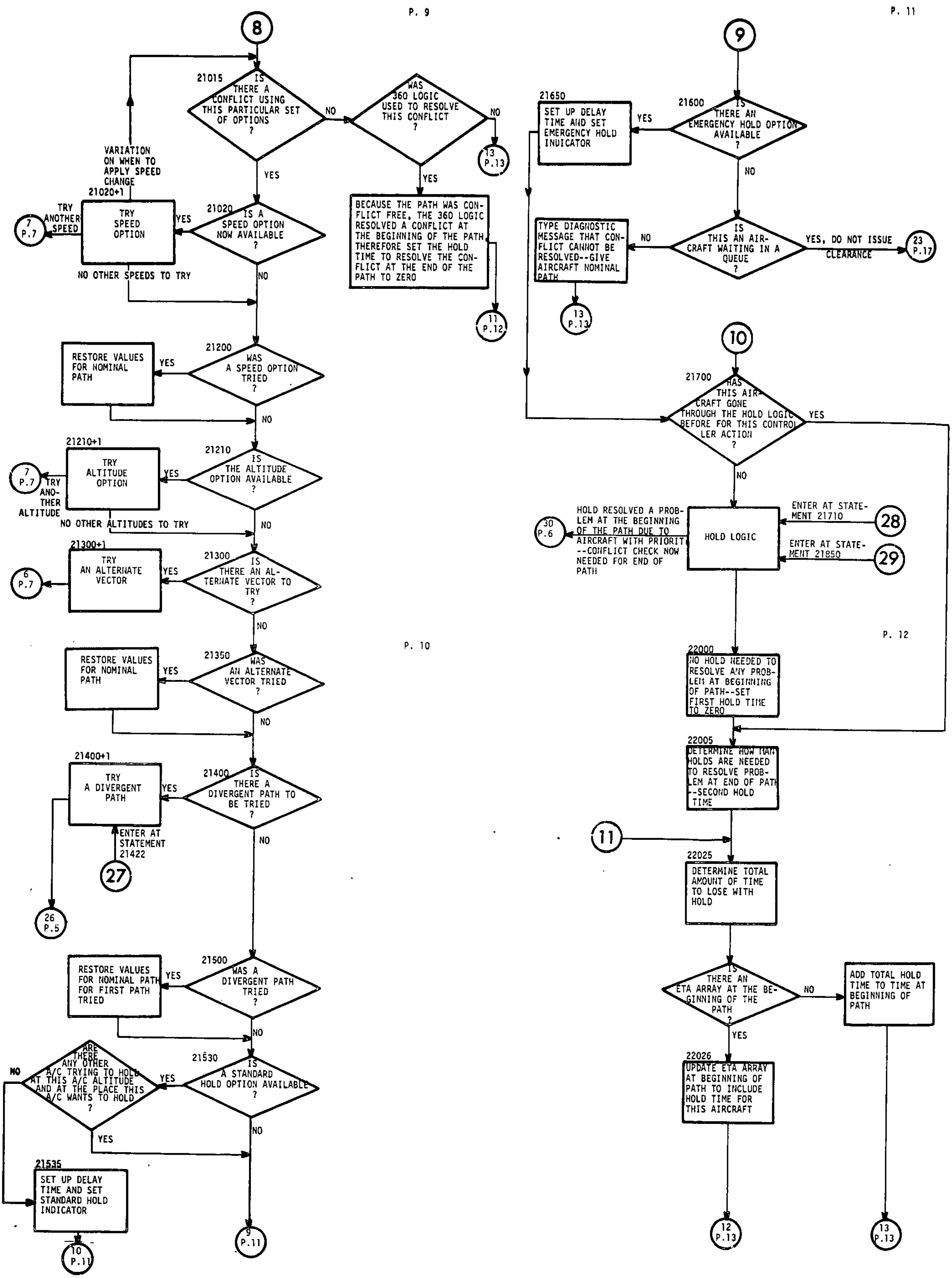


Fig. A-2.18. Continued.

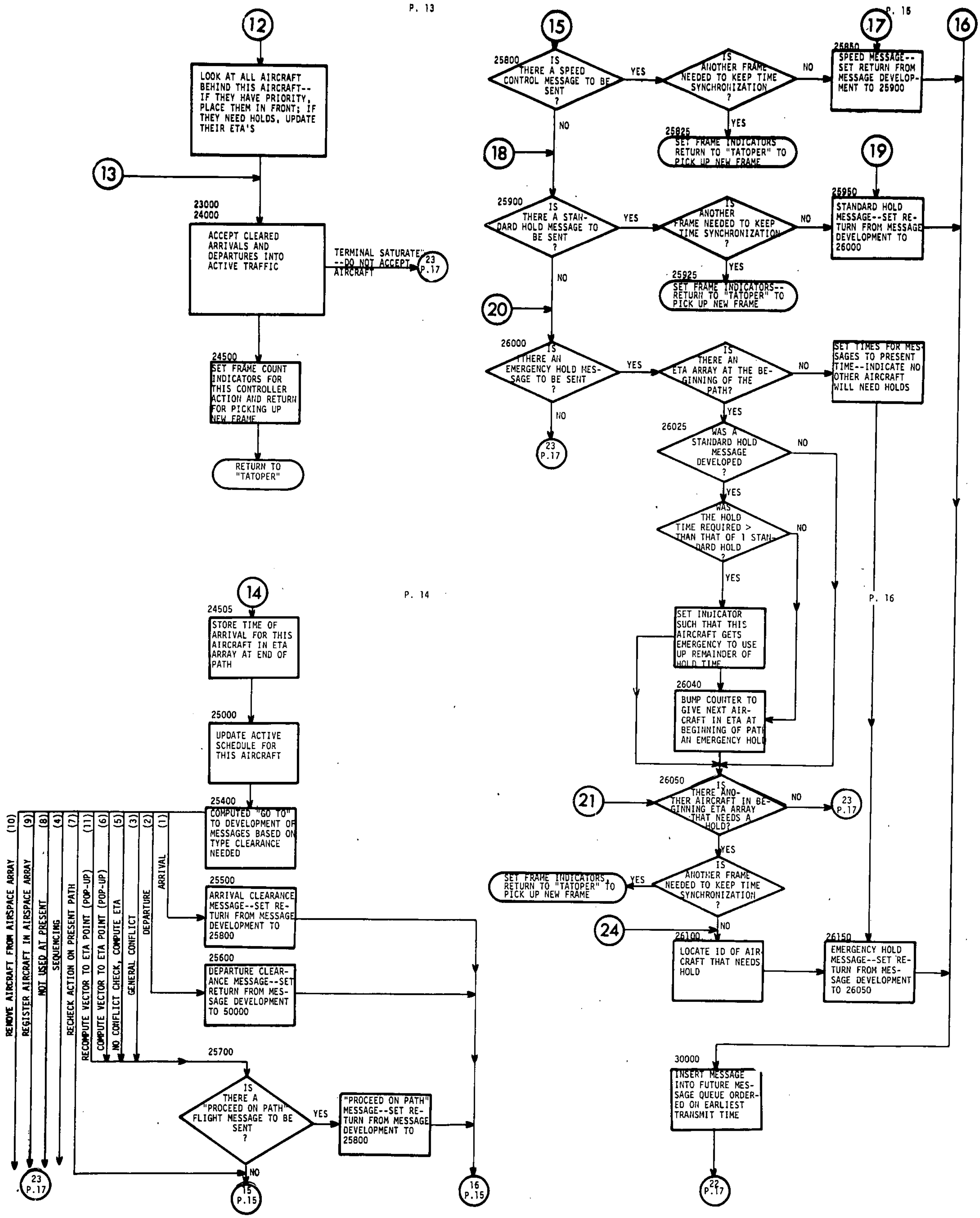


Fig. A-2.18. Continued.

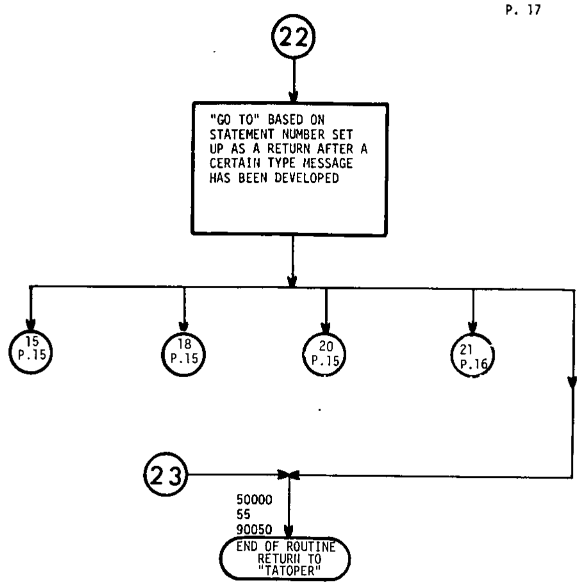


Fig. A-2.18. Concluded.

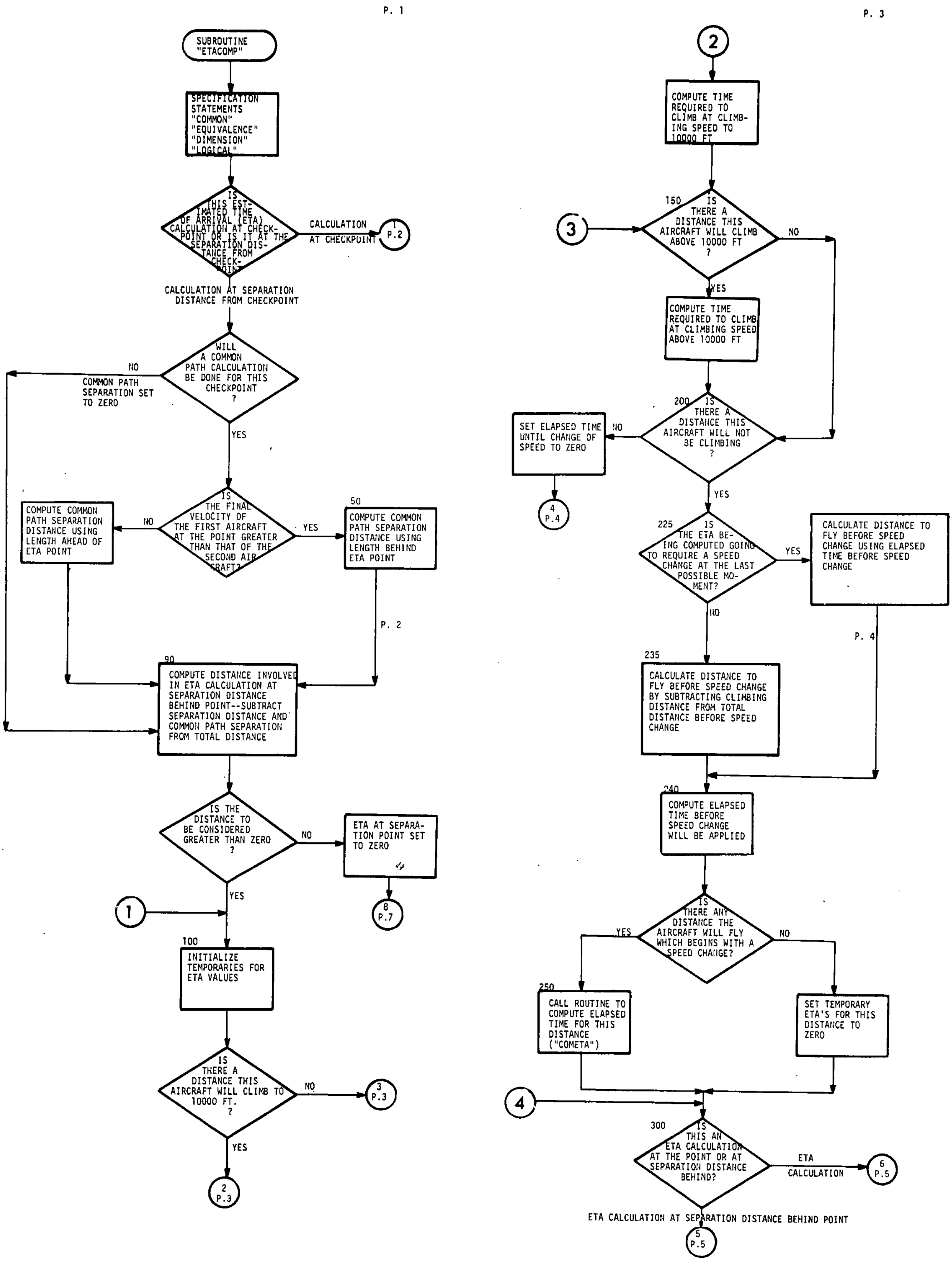


Fig. A-2.19. ETACOMP - compute ETA.

2

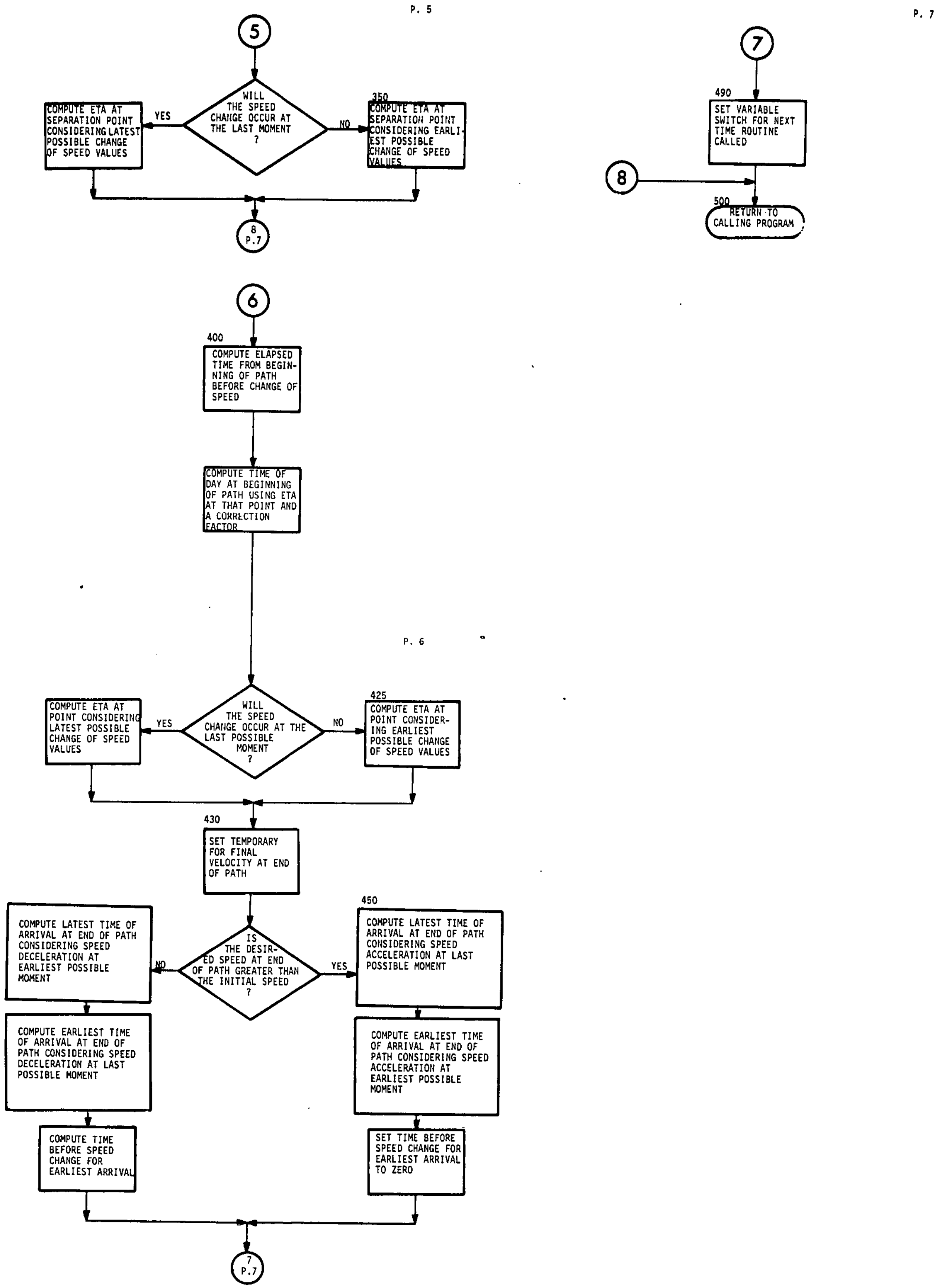


Fig. A-2.19. Concluded.

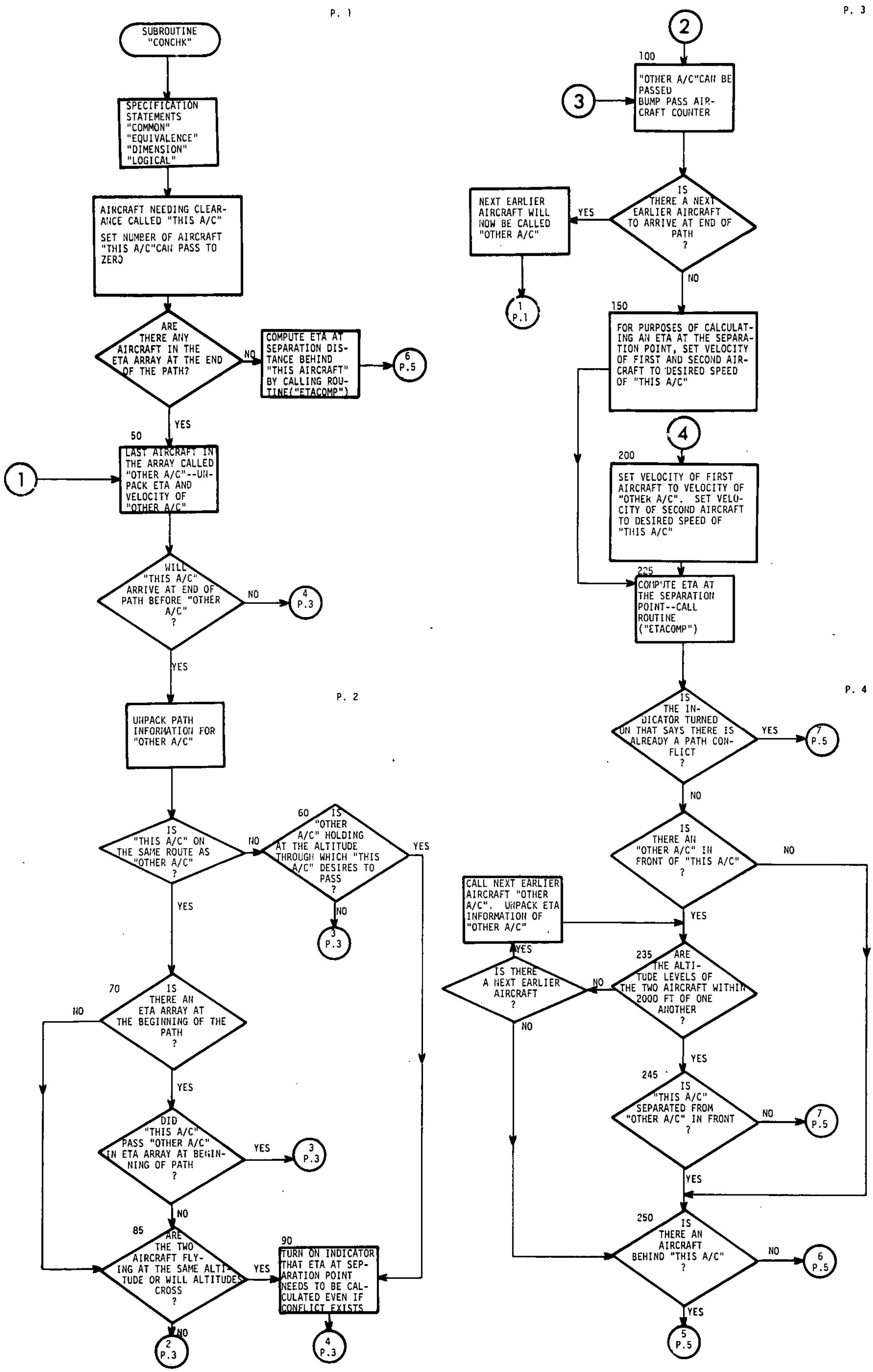


Fig. A-2.20. CONCHK - conflict check.

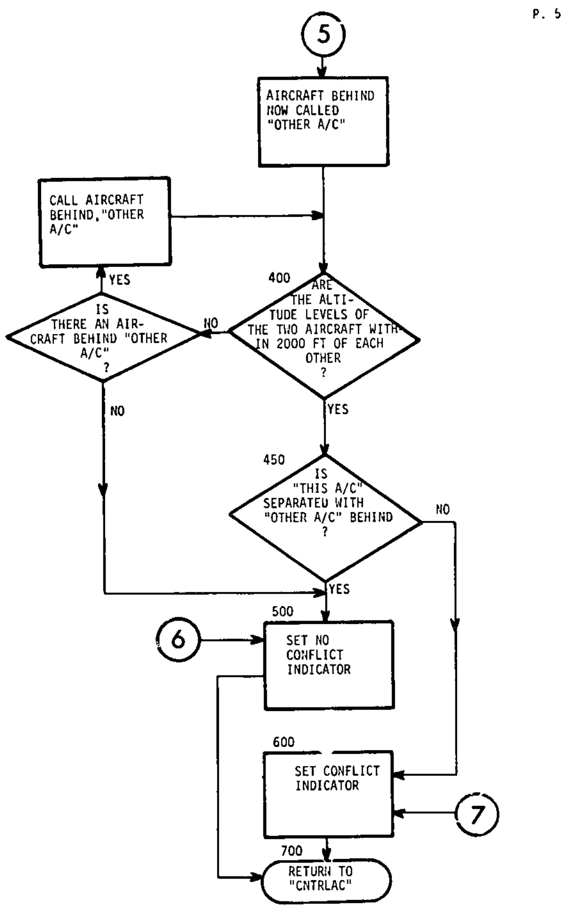


Fig. A-2.20. Concluded.

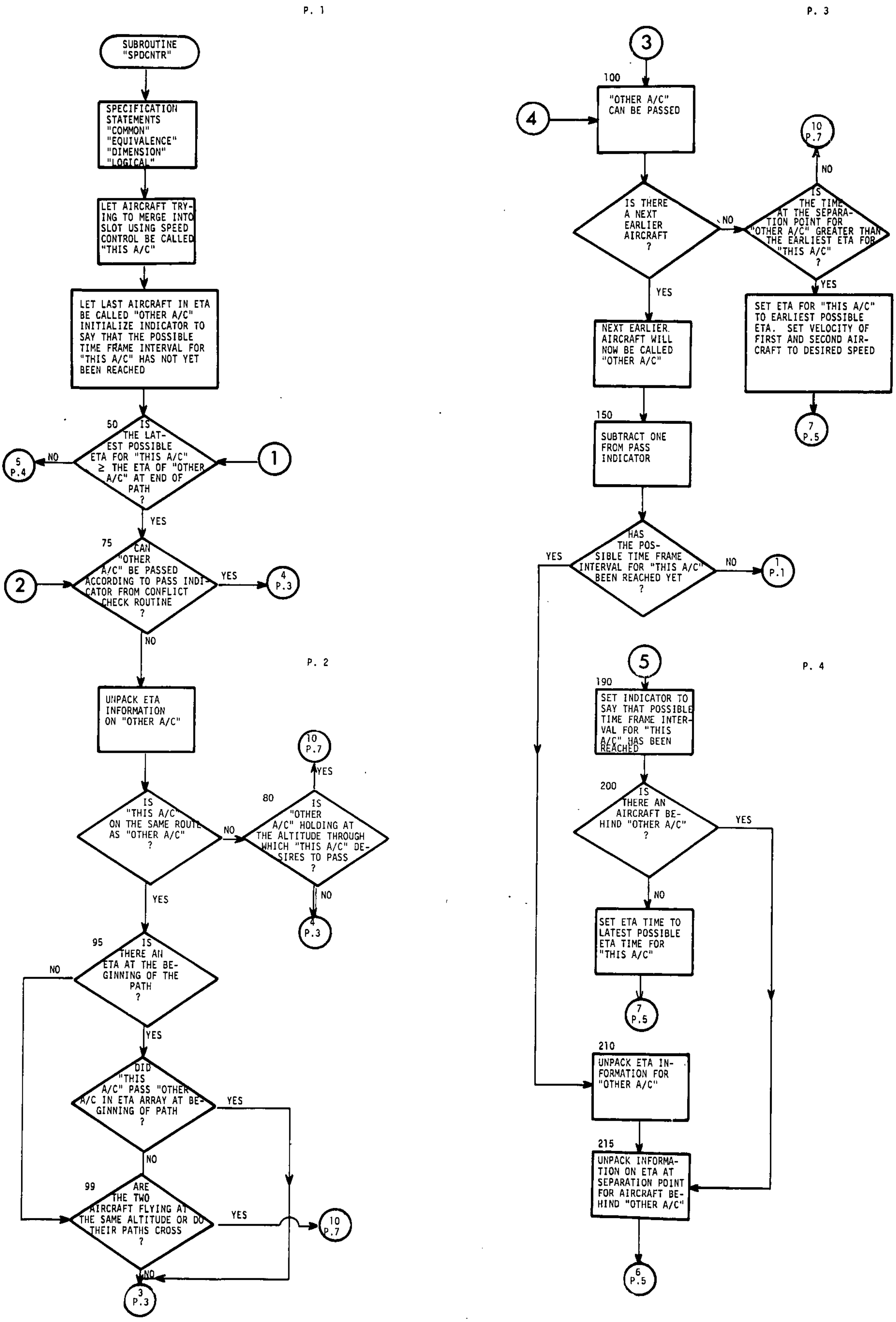


Fig. A-2.21. SPDCNTR - speed control.

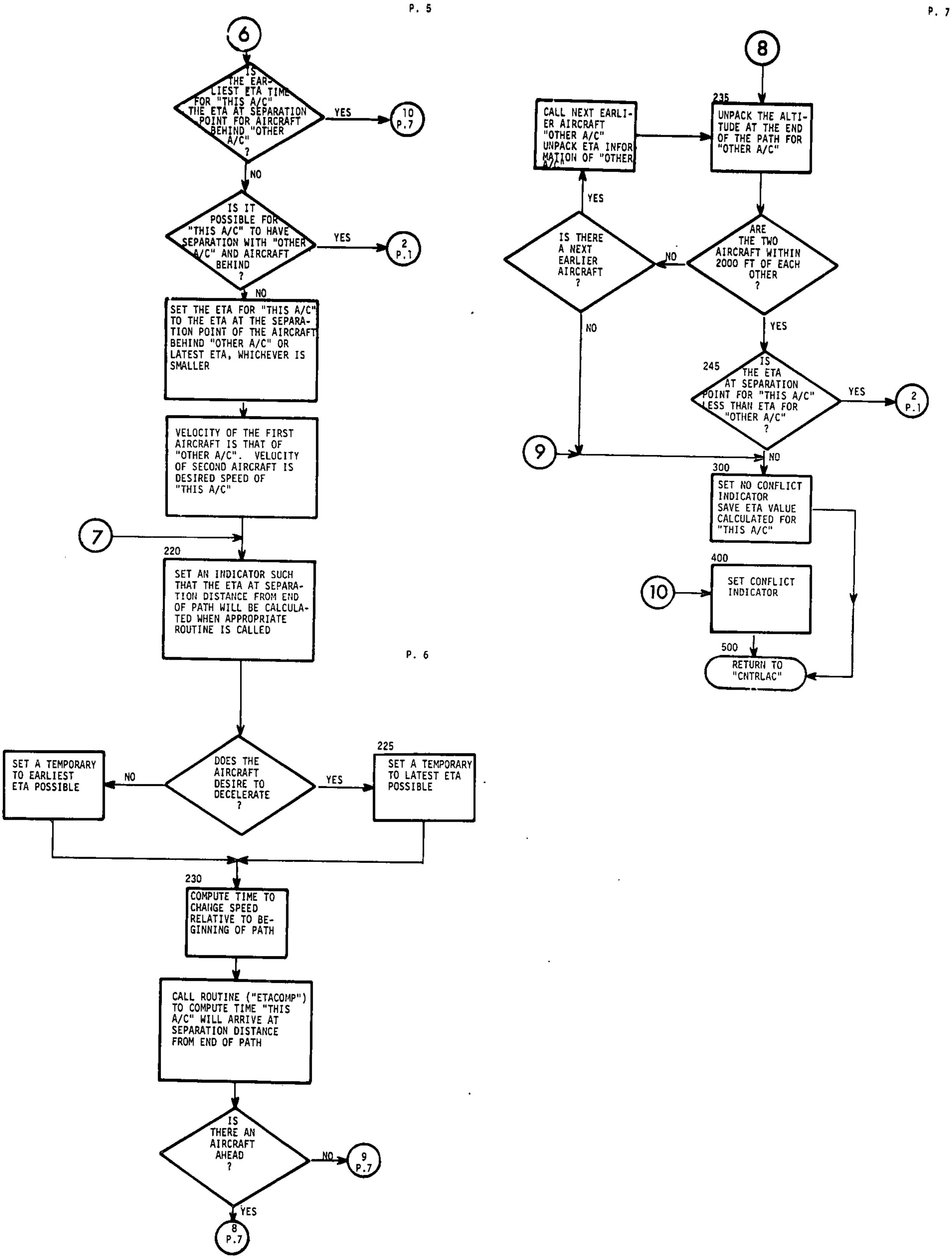


Fig. A-2.21. Concluded.

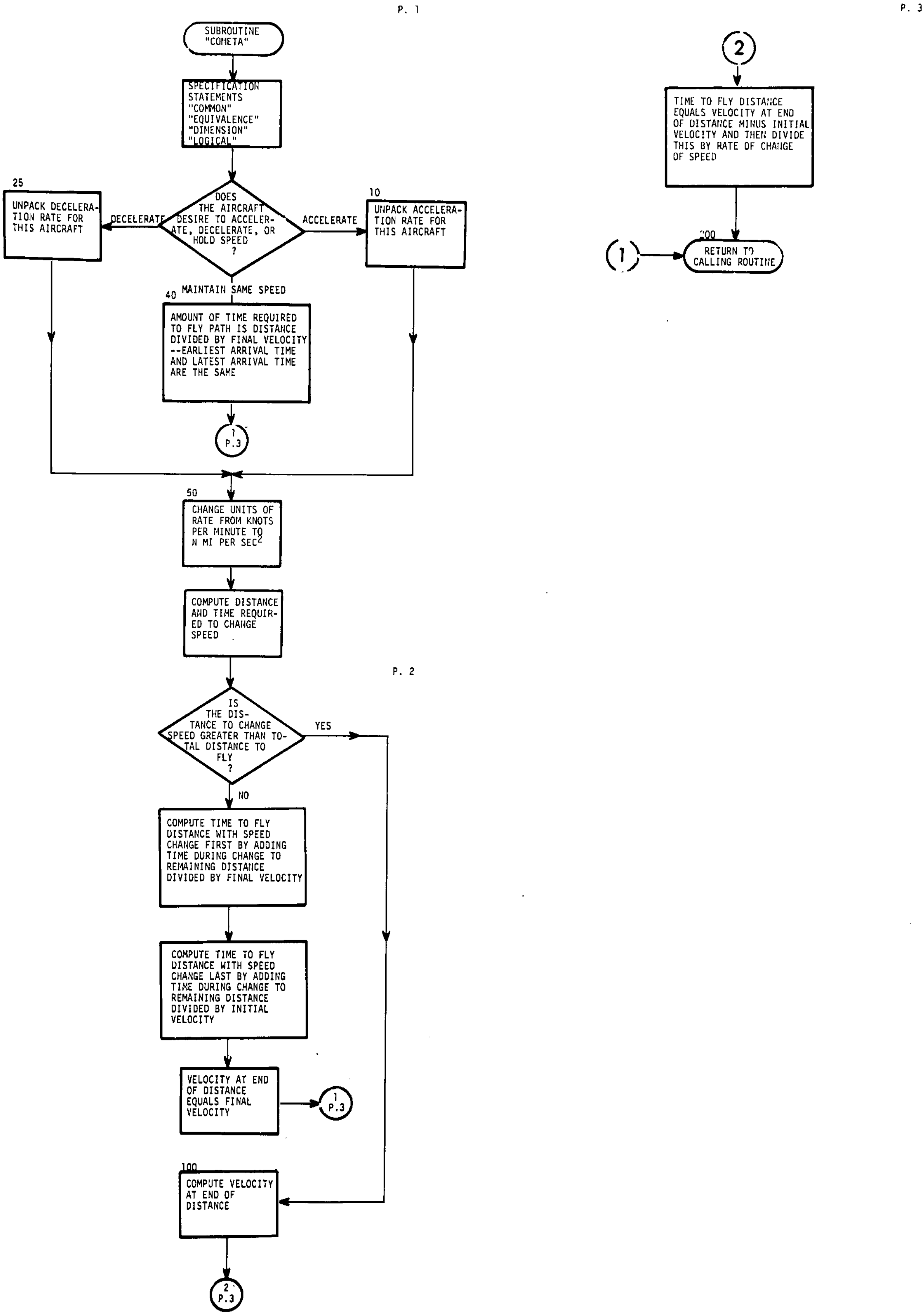


Fig. A-2.22. COMETA - compute time to fly a distance.

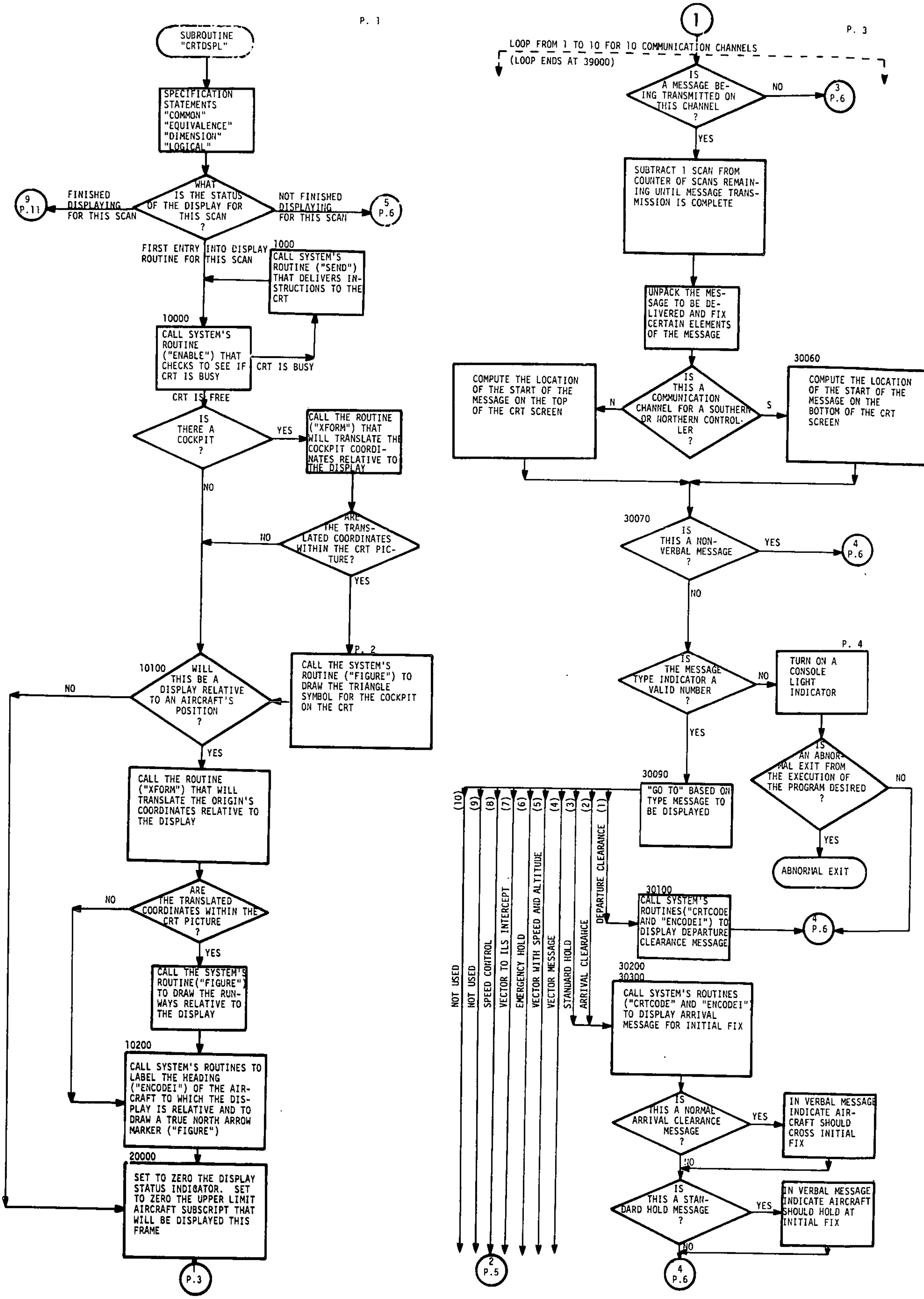


Fig. A-2.23. CRTDSPL - CRT display.

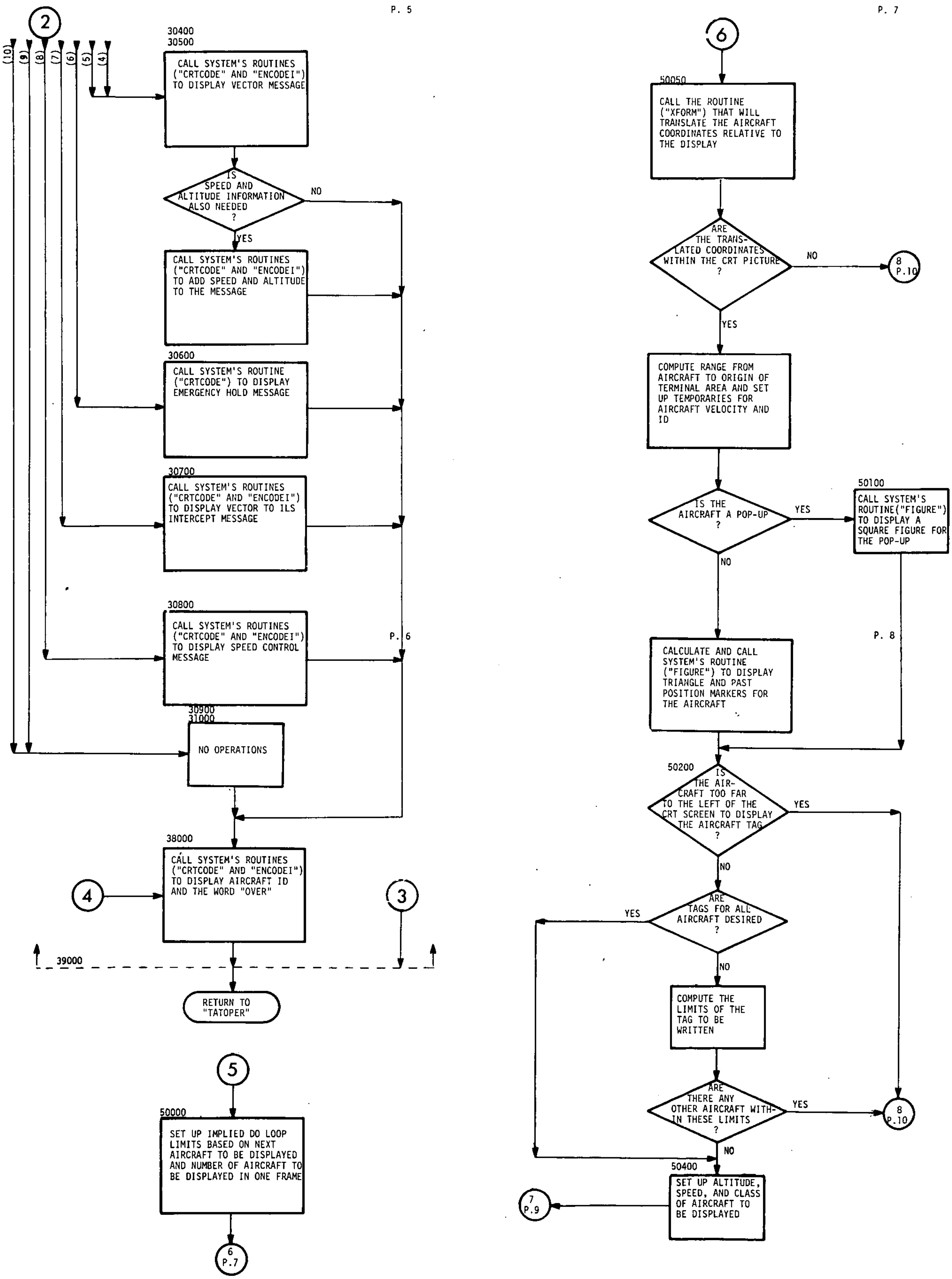


Fig. A-2.23. Continued.

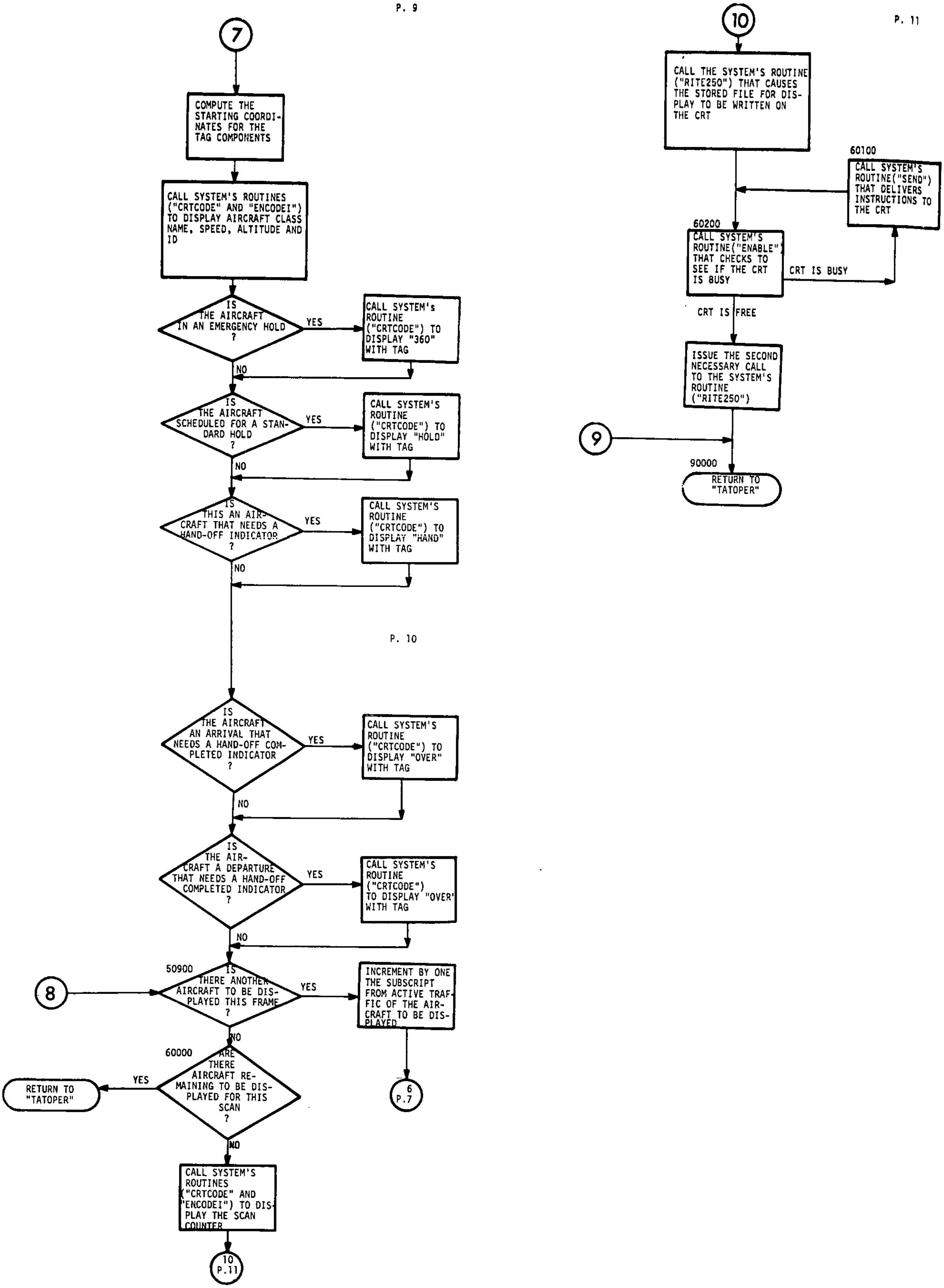


Fig. A-2.23. Concluded.

P. 1

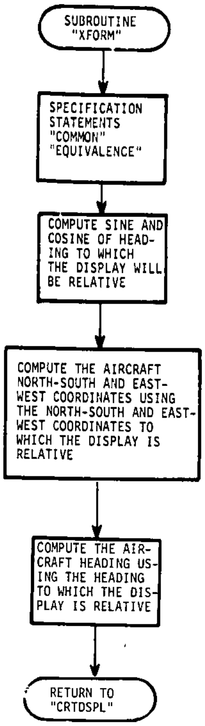


Fig. A-2.24. XFORM - coordinate transformation.

Fig. A-3. Logical flow charts for Traffic Generation Program.

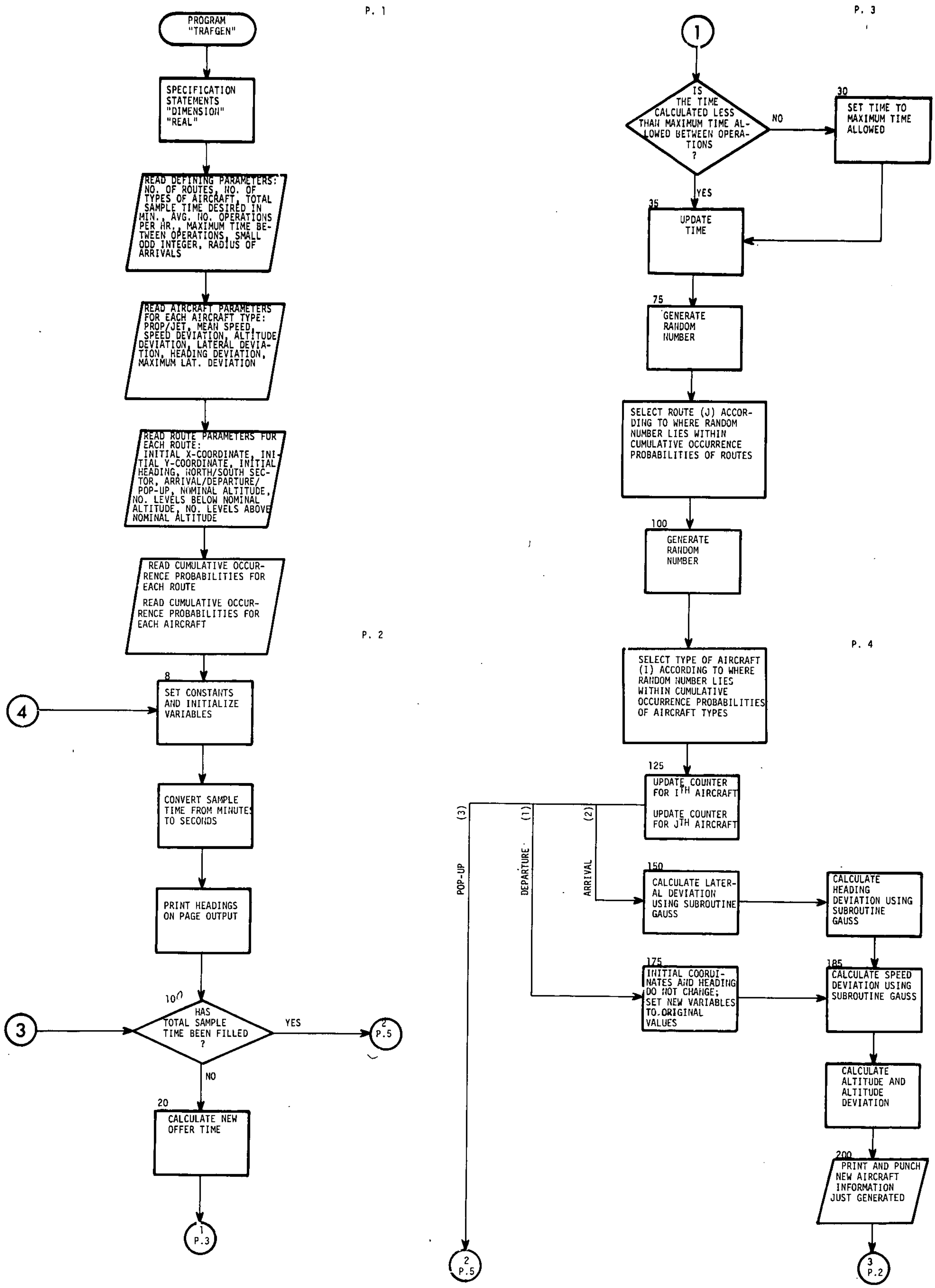


Fig. A-3.1. TRAFGEN - traffic generator.

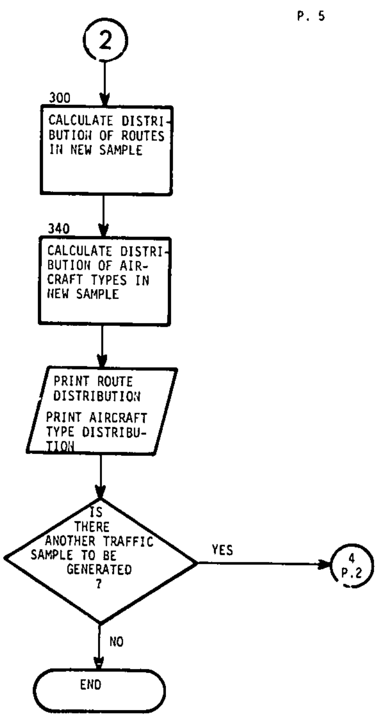


Fig. A-3.1. Concluded.

P. 1

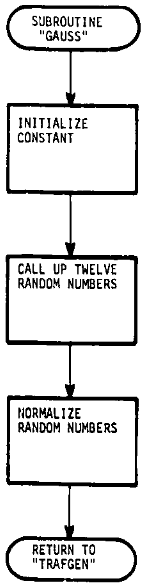


Fig. A-3.2. GAUSS - Gaussian.

Fig. A-4. Logical flow charts for the Analysis Program.

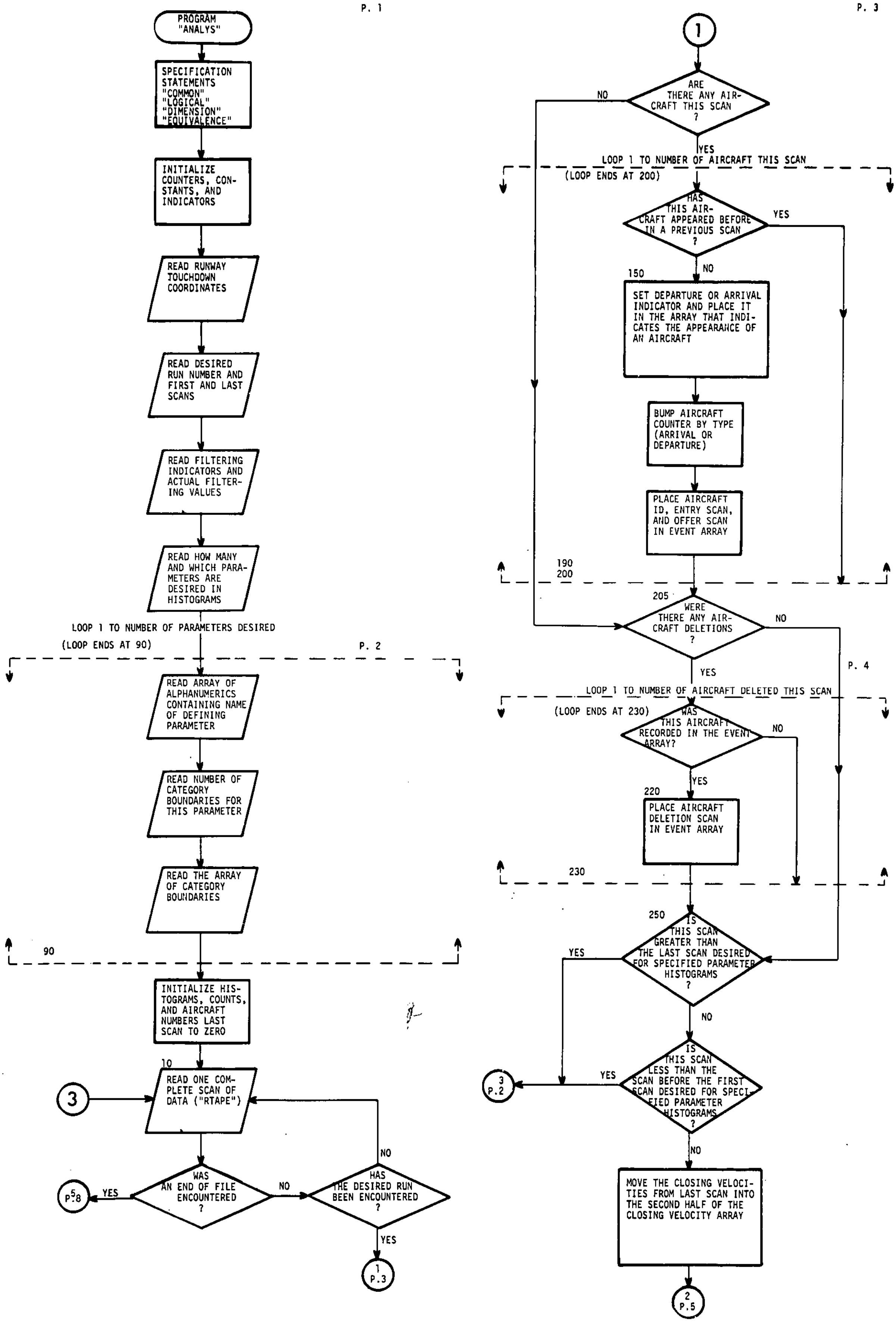


Fig. A-4.1. ANALYS - main program for analysis.

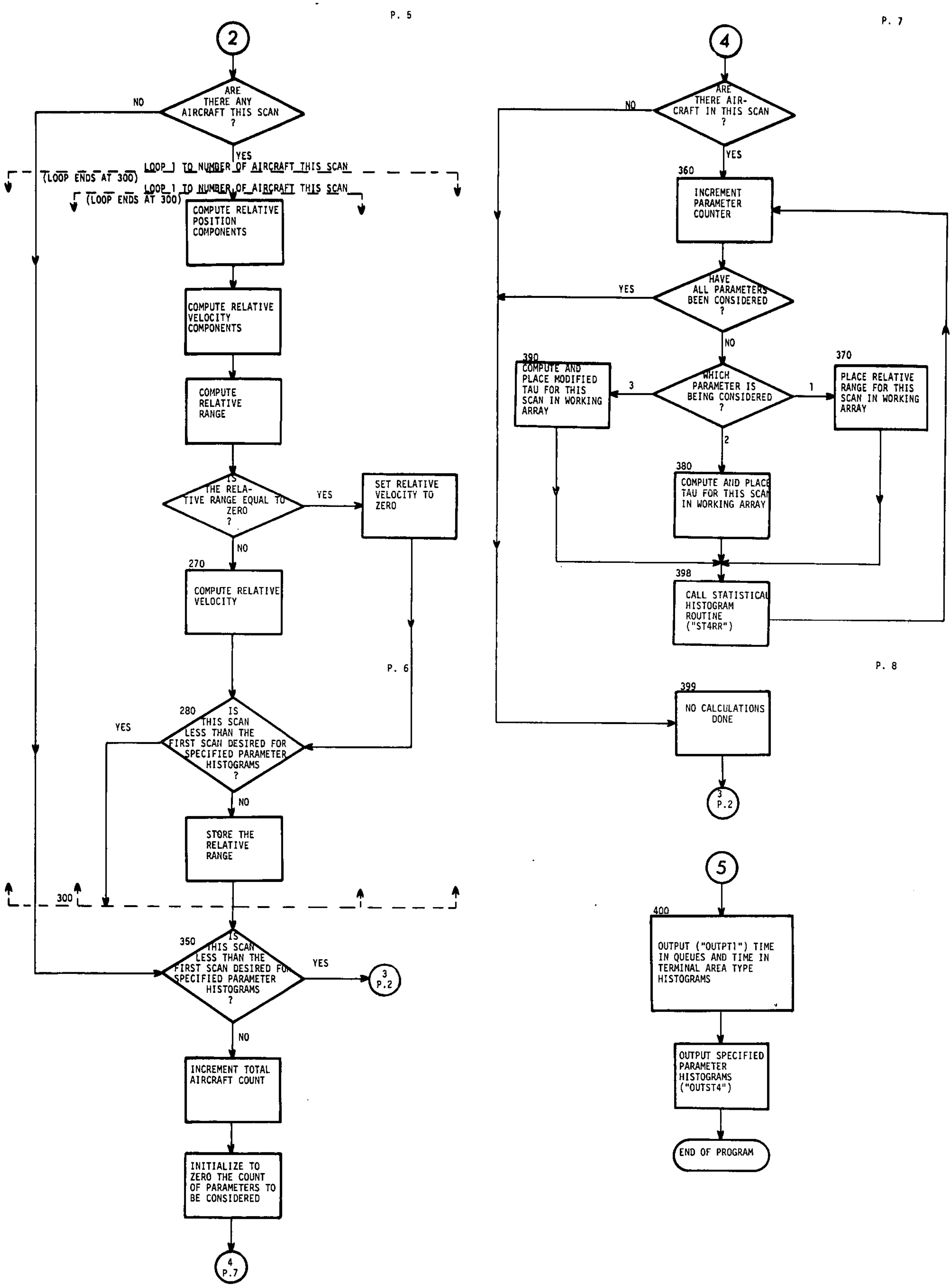


Fig. A-4.1. Concluded.

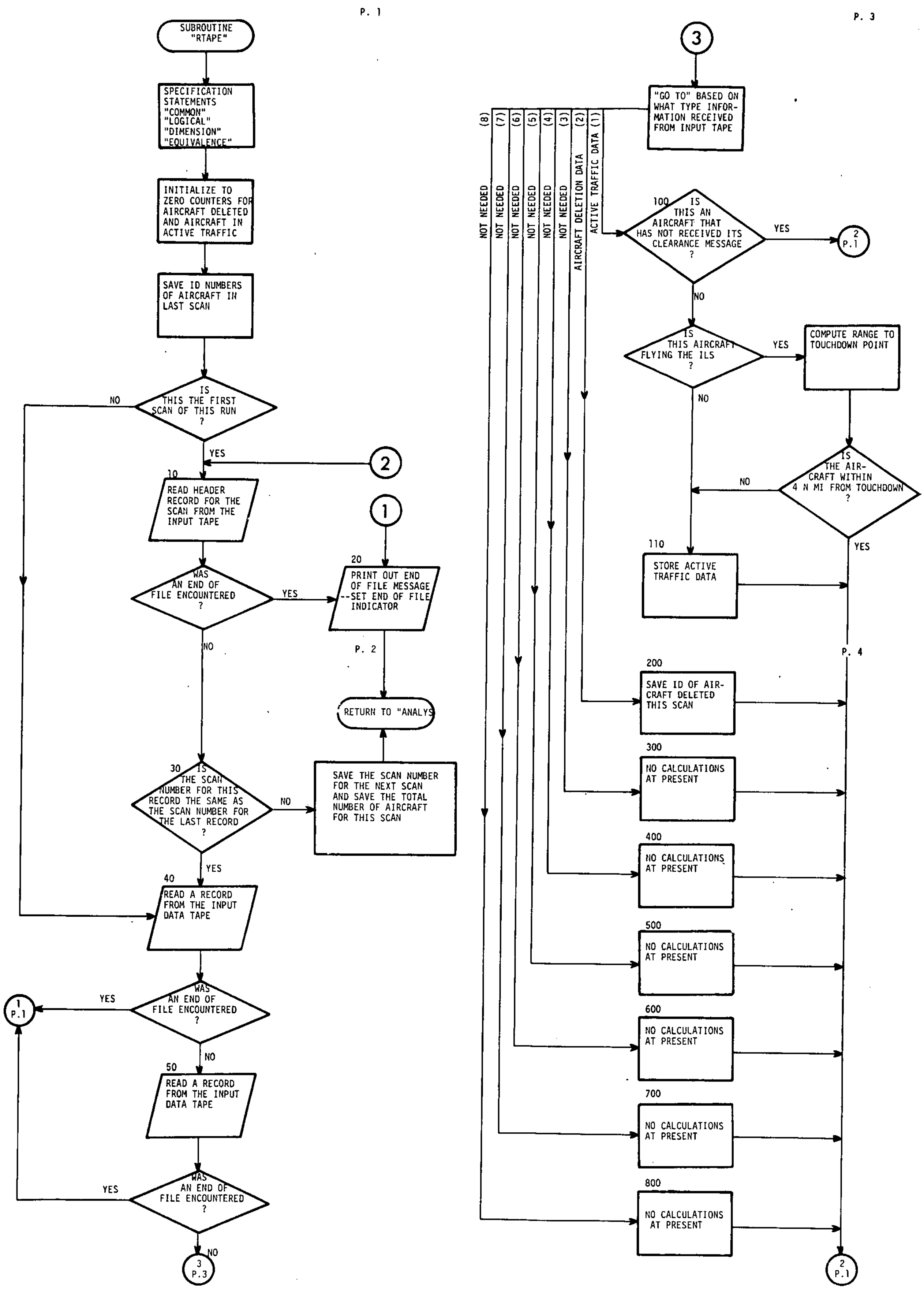


Fig. A-4.2. RTAPE - read tape.

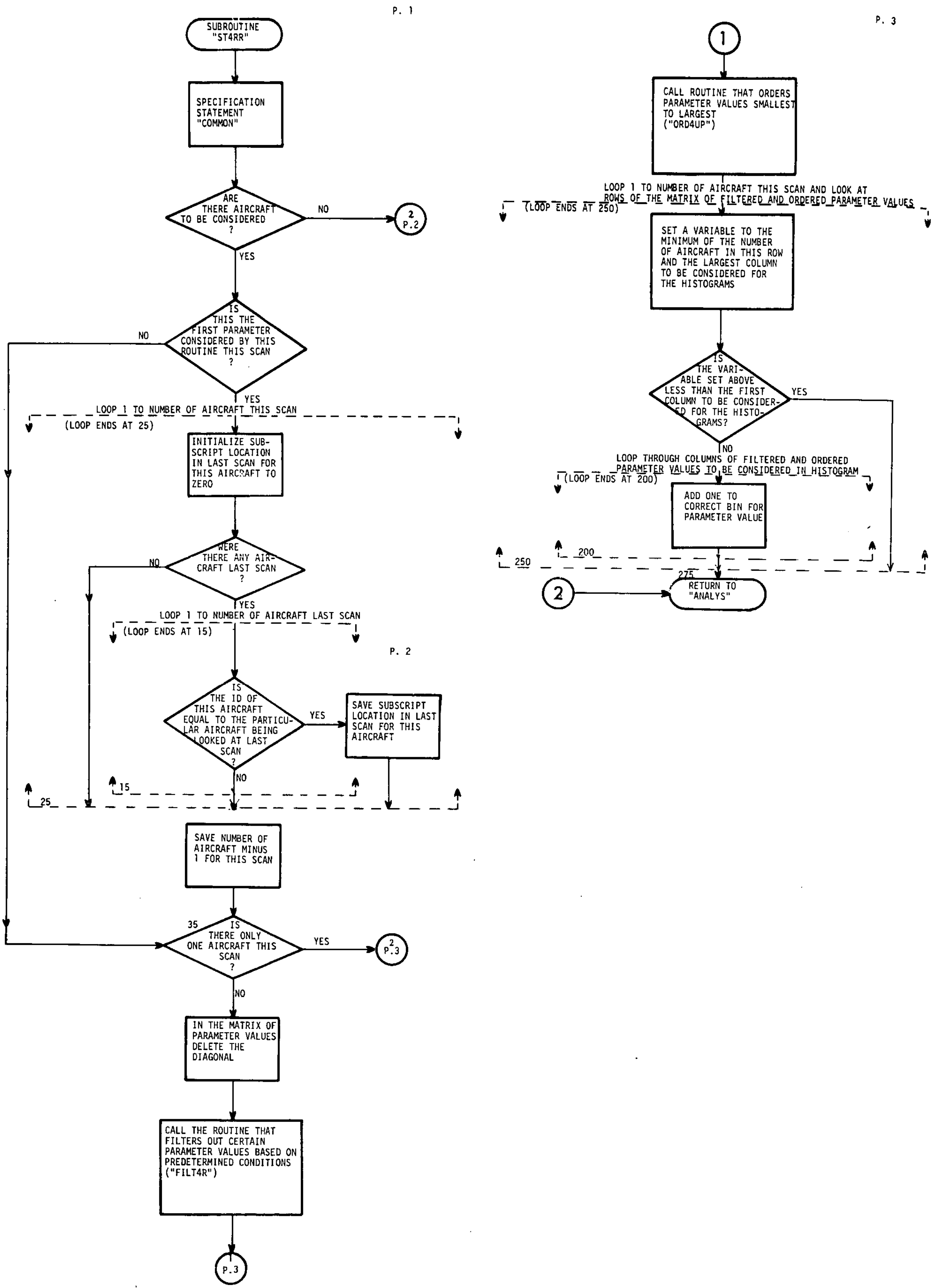


Fig. A-4.3. ST4RR - statistics of relationships between aircraft.

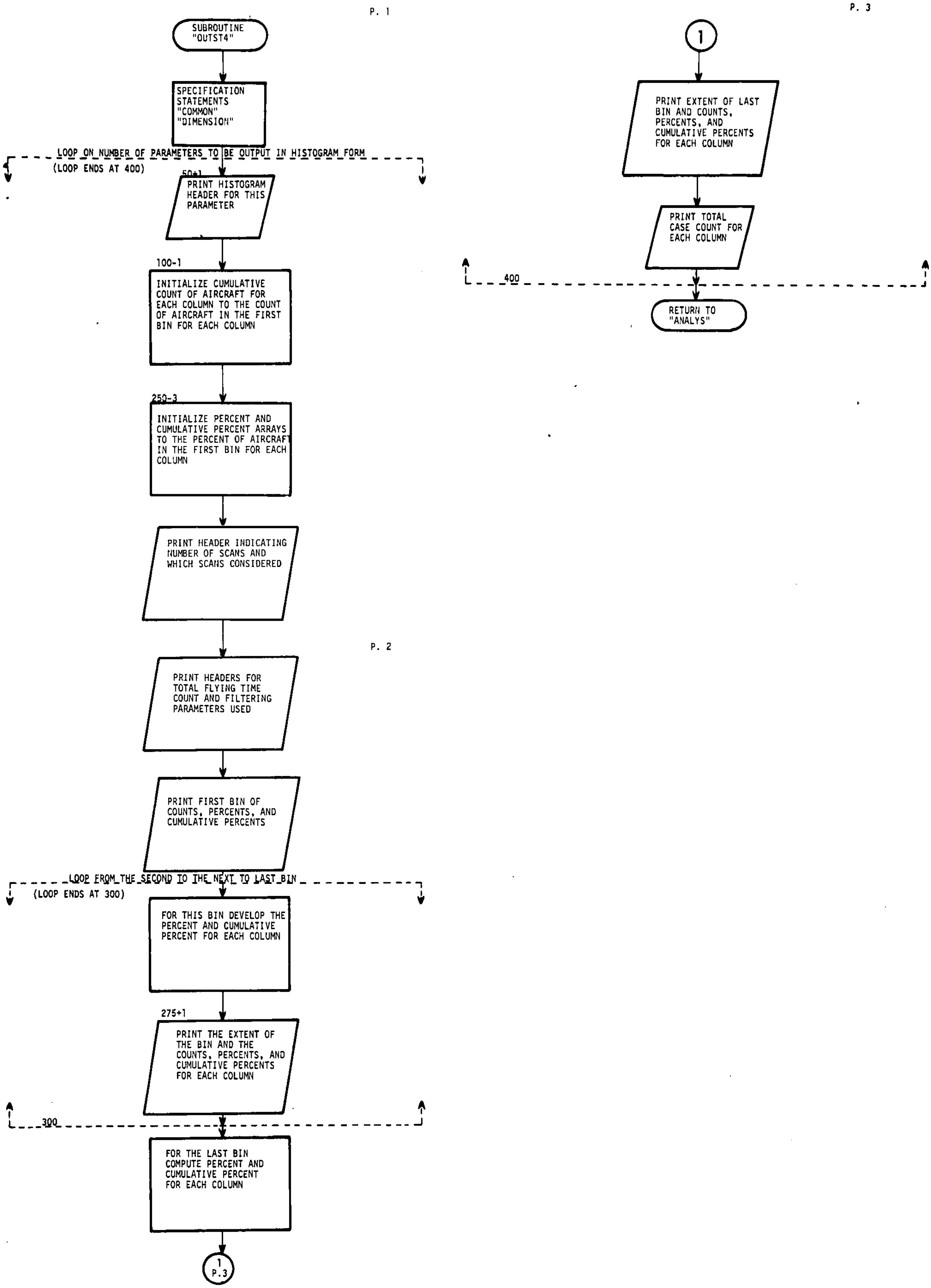


Fig. A-4.4. OUTST4 - output of ST4RR.

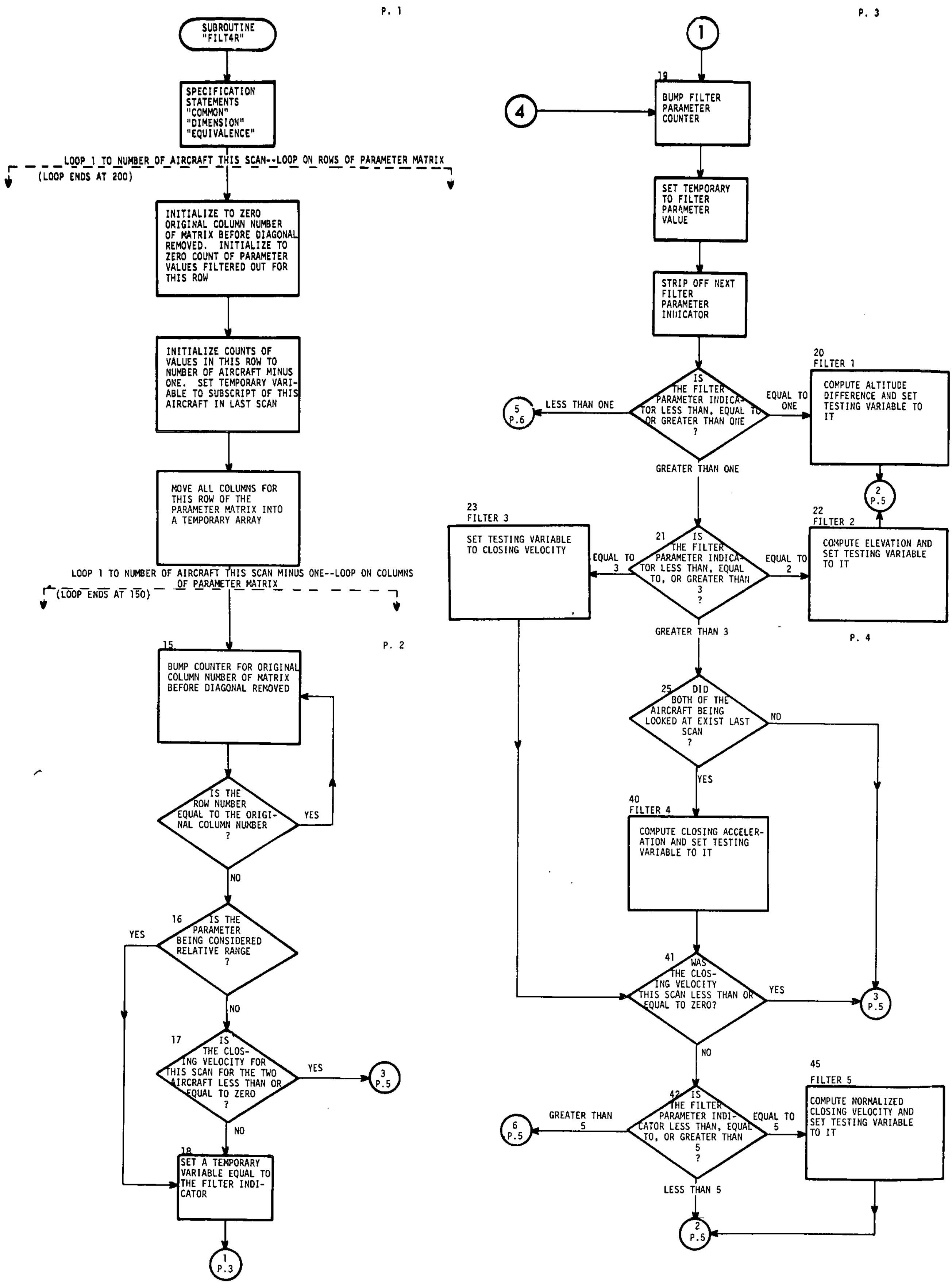


Fig. A-4.5. FILT4R - filtering of statistical relationships.

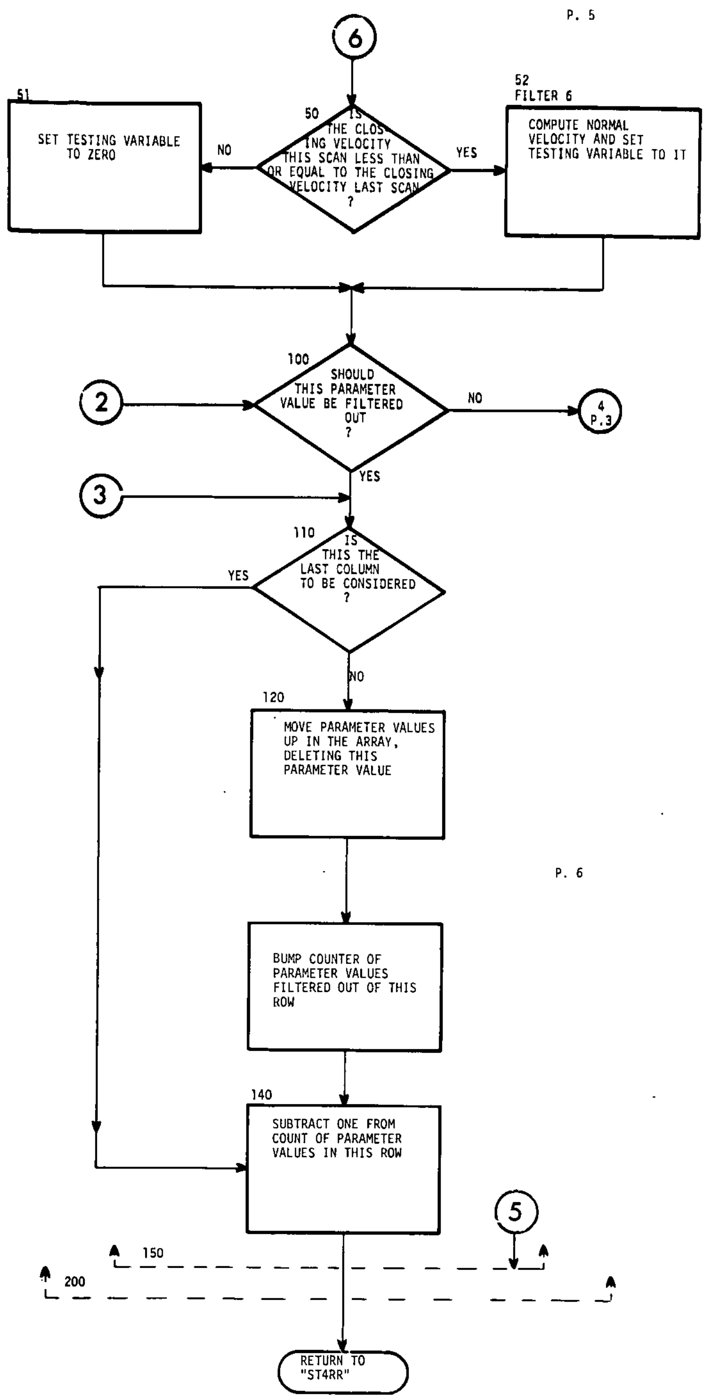


Fig. A-4.5. Concluded.

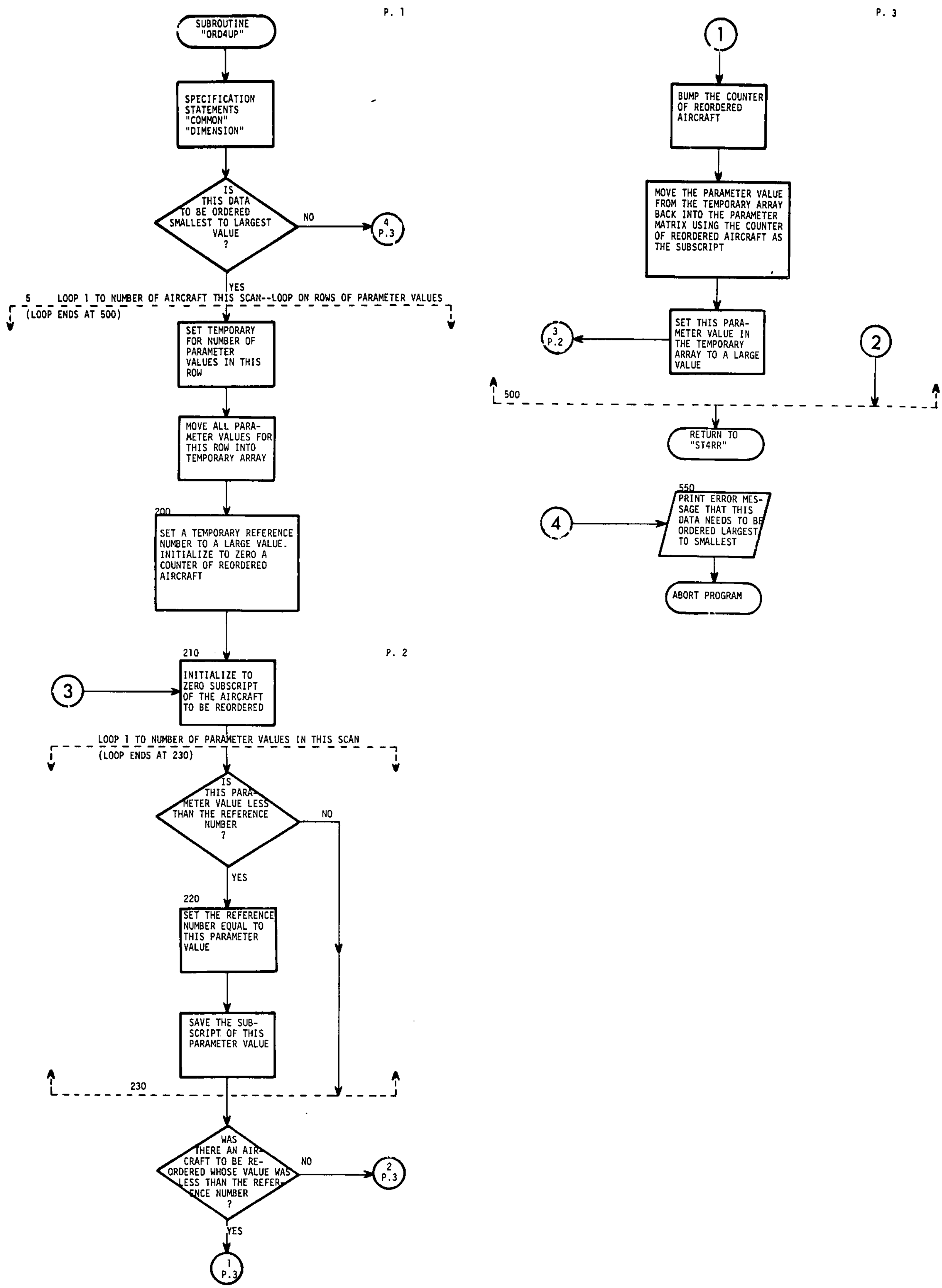


Fig. A-4.6. ORD4UP - array ordering.

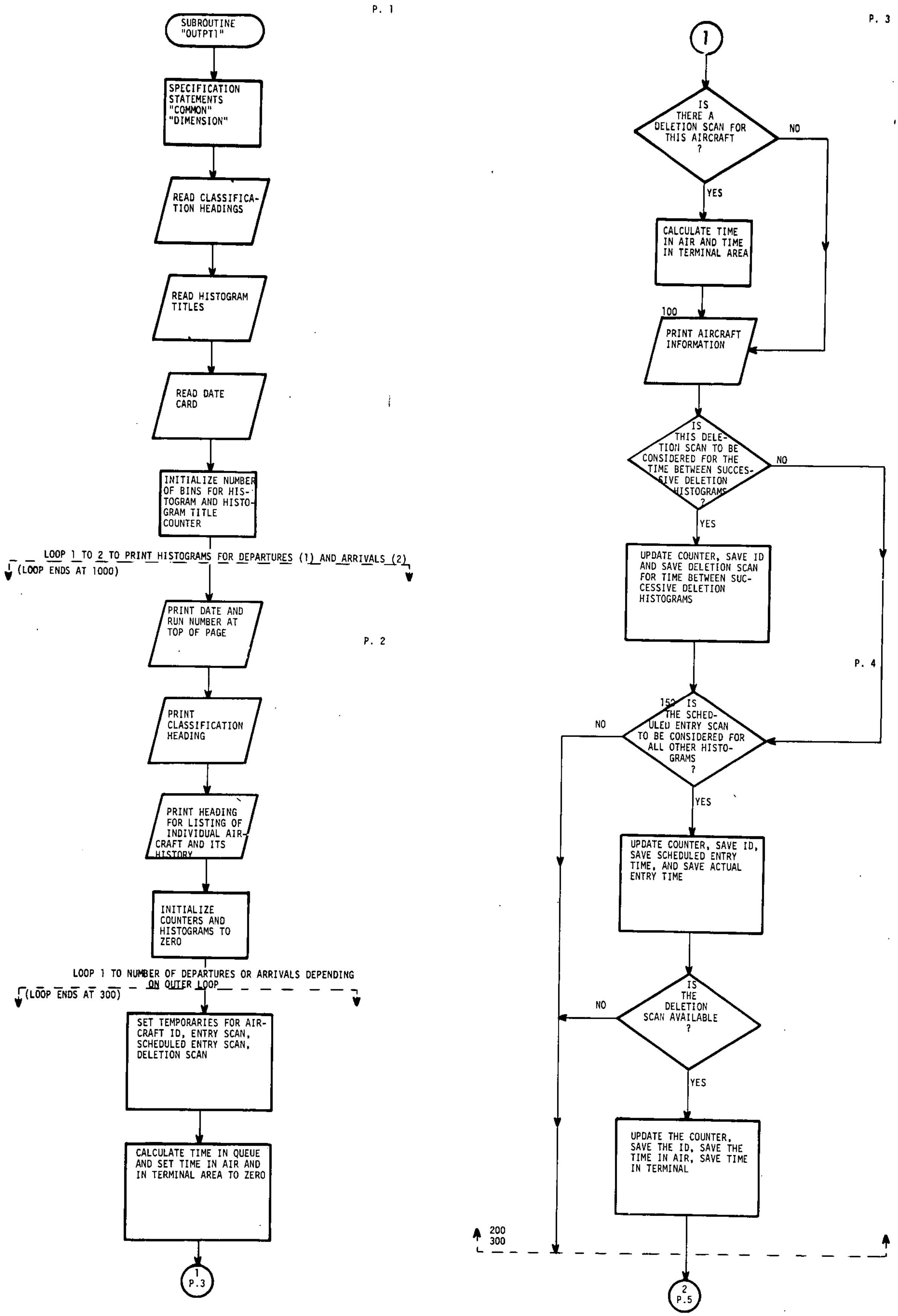


Fig. A-4.7. OUTPT1 - output of various duration statistics for aircraft in queues or in the terminal area.

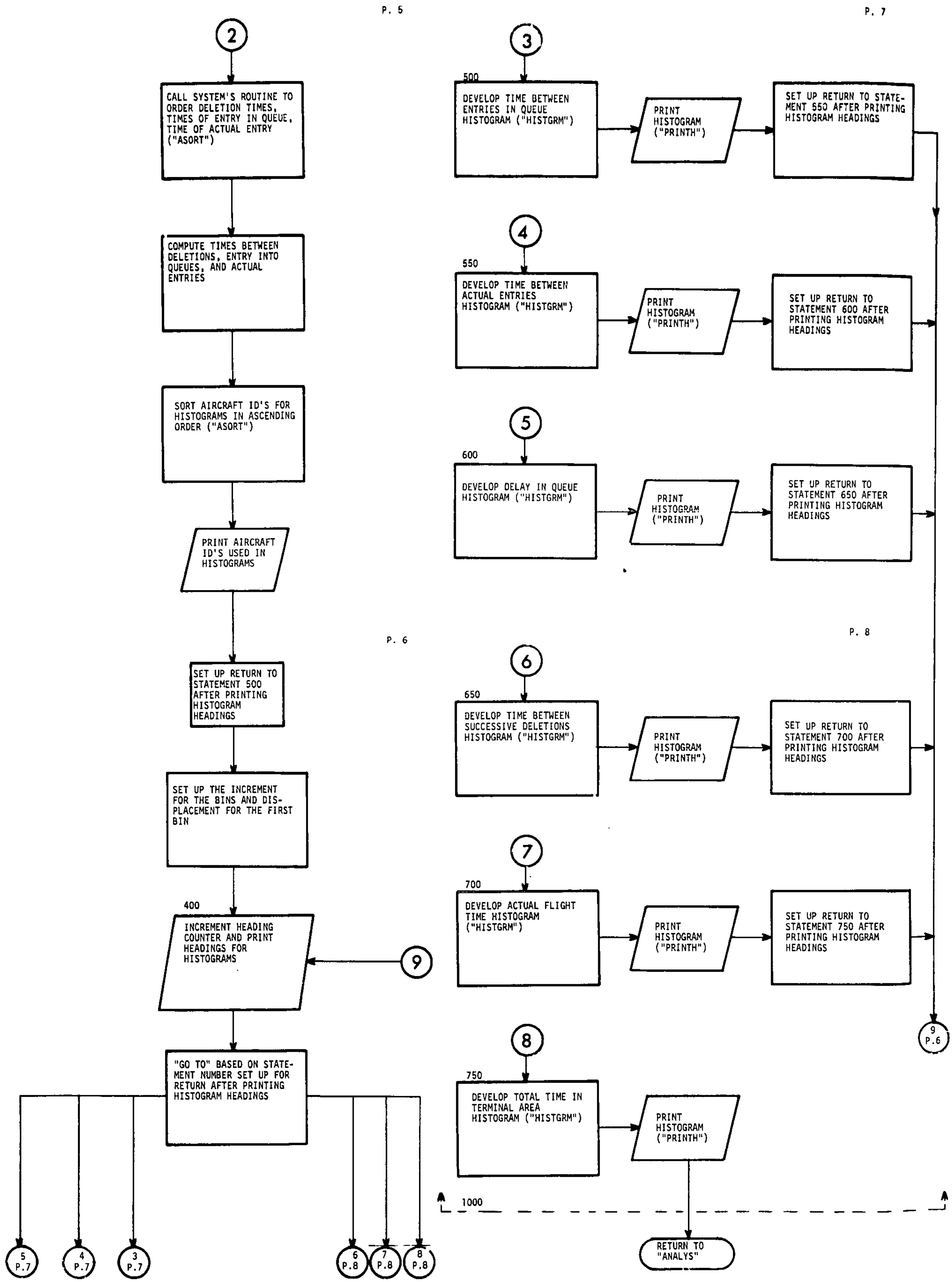


Fig. A-4.7. Concluded.

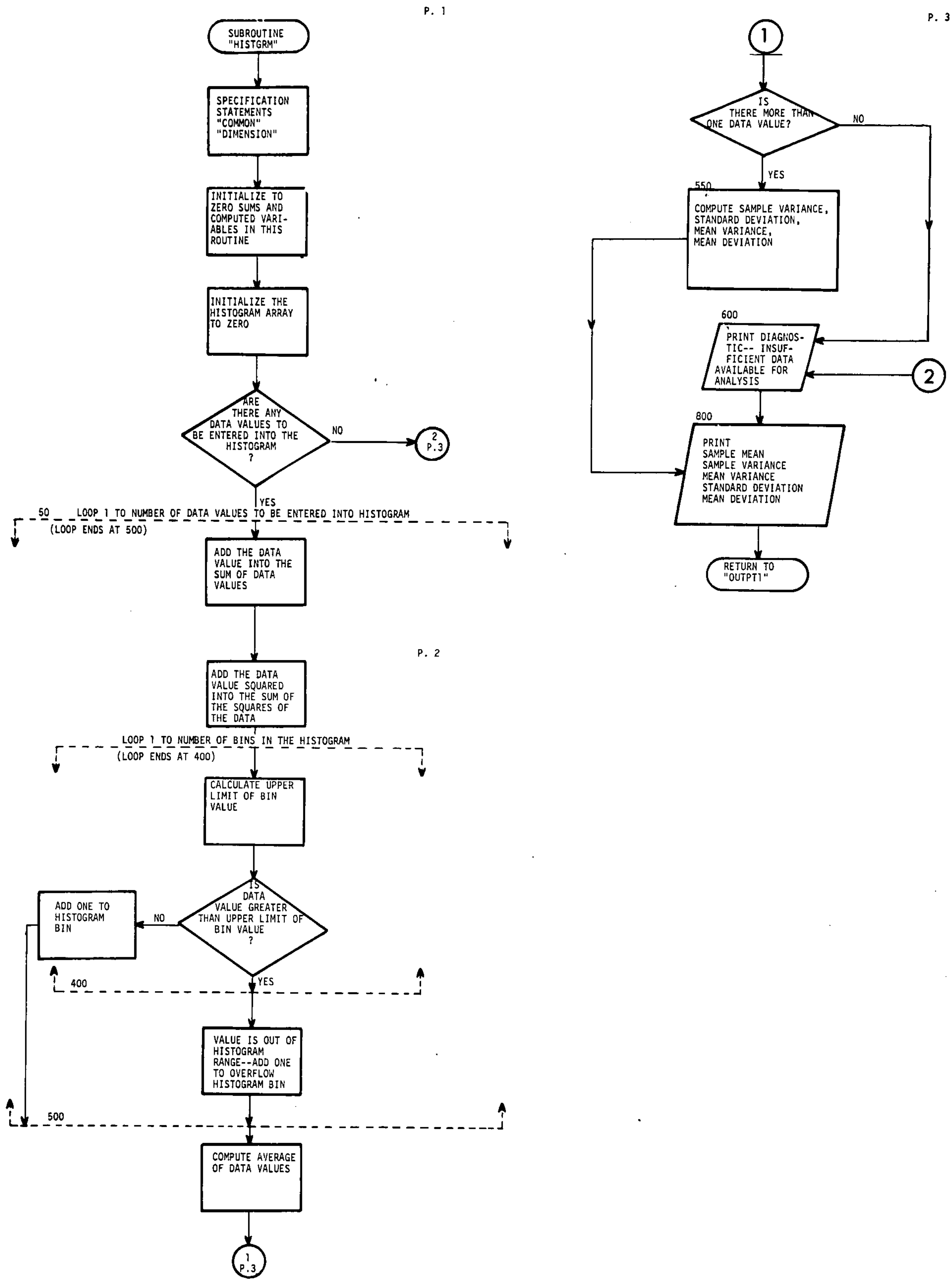


Fig. A-4.8. HISTGRM - histogram.

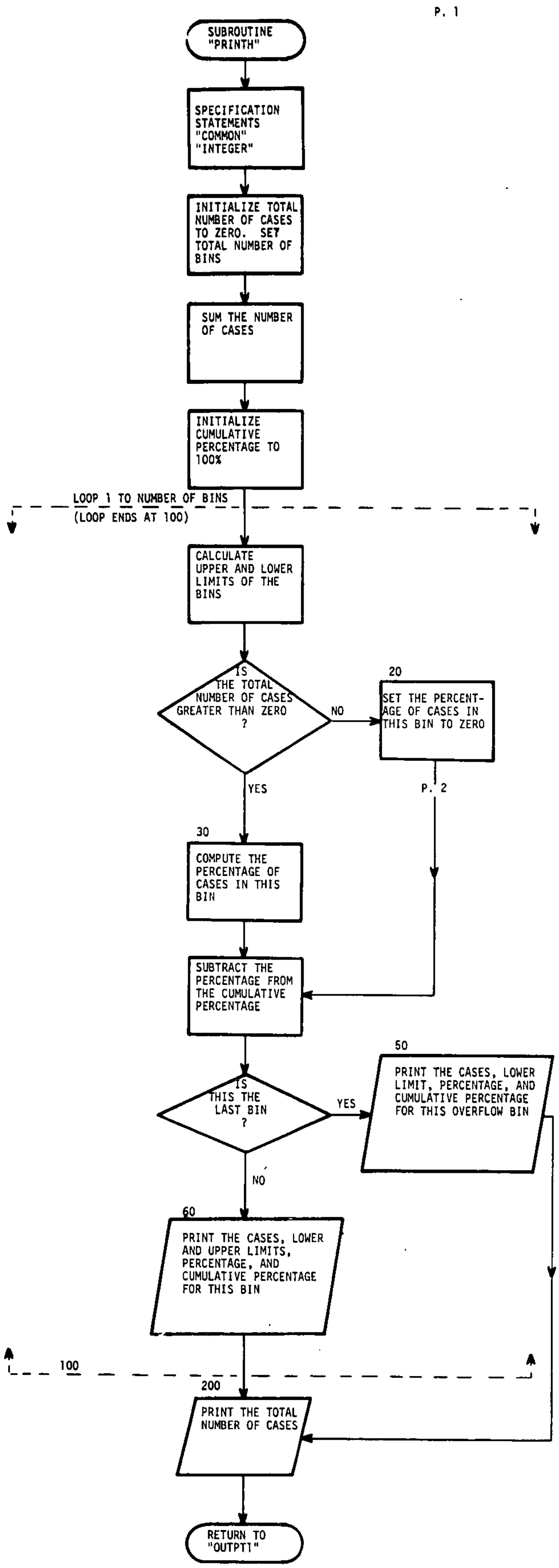


Fig. A-4.9. PRINTH - print histogram.

APPENDIX B

Input Data for the Current Atlanta Terminal Area Route Structure

This appendix contains filled-out data sheets giving the set of numerical values used to define the current Atlanta route structure and procedures. In addition, data sheets which define and list the aircraft performance parameters currently in use in the simulation are documented in this appendix. Table B-1 defines and gives current numerical values for miscellaneous input parameters that are not included in the filled-out data sheets.

Page 1 of 2 AIRCRAFT PERFORMANCE Date 8/8/72

	(IPERFR - Packed Array)							(IPERFS - Packed Array)								(IPERFT(I))		(IPERFTR(I))
	Climb/Descent Rates (ft/min x 10^{-2})							Speeds (Kts)										
A/C Type I	Maximum Descent	Enroute Descent	Terminal Descent	Climb to 10,000 ft	Climb to 20,000 ft	Acceleration (Kts/min)	Deceleration (Kts/min	Take-off	Climb to 10,000 ft	Climb to 20,000 ft	Cruising	Transi-tion	Terminal	Approach	Final	Seconds to Lift-off (dep.)	Runway Occupancy Time (arr.)	Standard Rate of Turn
1	5000	5000	1500	4000	4000	65	60	120	250	270	460	250	200	160	125	30		3
2	5000	5000	1500	4000	4000	65	60	120	250	270	460	250	200	160	125	30		3
3	5000	5000	1500	4000	4000	70	60	120	250	275	465	250	200	150	125	30		3
4	5000	5000	1500	4000	4000	70	60	120	250	275	450	250	200	150	125	30		3
5	5000	5000	1500	4000	4000	65	60	120	250	270	470	250	200	160	125	30		3
6	5000	4000	1500	3500	3500	60	60	125	250	265	480	250	200	155	130	30		3
7	3000	2000	1500	2000	1000	60	60	80	135	125	140	140	140	140	120	40		3
8	3000	2000	1500	2000	1500	70	60	100	145	135	165	165	165	120	105	35		3
9	4000	2500	1500	2500	2500	80	60	100	160	165	200	200	200	135	110	30		3
10	3000	2000	1500	2000	1000	60	60	80	135	125	140	140	140	115	85	40		3
11	5000	4000	1500	3500	3500	60	60	125	250	265	250	250	200	155	130	30		3
12	3000	2000	1500	2000	1500	70	60	100	145	135	165	165	165	120	105	35		3
13	3000	2000	1500	2000	2000	80	60	105	175	170	200	200	200	135	110	30		3
14	4000	2500	1500	2500	2500	80	60	100	160	165	200	200	200	135	110	30		3
15	5000	5000	1500	2500	2500	70	60	125	250	[illegible]	250	250	200	165	125	30		3
16	5000	5000	1500	2500	2500	70	60	125	250	[illegible]	250	250	200	165	125	30		3
17	3000	2000	1500	2000	2000	80	60	105	175	170	200	200	200	135	110	30		3
18	5000	5000	1500	3000	3000	60	60	125	250	255	250	250	200	160	125	32		3

AIRCRAFT PERFORMANCE

Date 8/8/72

	(IPERFR - Packed Array)							(IPERFS - Packed Array)								(IPERFT(I))		(IPERFTR(I))
	Climb/Descent Rates (ft/min x 10^{-2})							Speeds (Kts)										
A/C Type I	Maximum Descent	Enroute Descent	Terminal Descent	Climb to 10,000 ft	Climb to 20,000 ft	Acceleration (Kts/min)	Deceleration (Kts/min	Take-off	Climb to 10,000 ft	Climb to 20,000 ft	Cruising	Transi-tion	Terminal	Approach	Final	Seconds to Lift-off (dep.)	Runway Occupancy Time (arr.)	Standard Rate of Turn
19	5000	5000	1500	3000	3000	60	60	125	250	255	485	250	200	160	125	32		3
20	5000	2500	1500	3000	3000	50	60	115	240	240	250	250	200	140	120	35		3
21	1000	1000	1000	700	500	23	86	35	60	60	90	90	90	50	50	7		1.5

Date 8/8/72

NAVIGATIONAL AIDS

	X Coord. (n mi)	Y Coord. (n mi)	Instrument Error (deg rms-n mi rms for DME)	Make Good radius (n mi)
ADF ID				
BRL 1	.25	5.96	2.0	.5
LSM 2	18.80	-12.44	2.0	.5
VOR ID				
ATL 1	-8.05	-2.42	1.3	1.
REG 2	.20	8.45	1.3	1.
OCR 3	18.51	15.30	1.3	1.
KMG 4	31.25	-34.09	1.3	1.
CSG 5	-61 58	-28.30	1.3	1.
9R 6	-.38	.84	1.3	1.
9L 7	.34	1.40	1.3	1.
FTY 8	8.05	-4.26	1.3	1.
FIX ID				
ATL 1	-8 05	-2.42	.5	.5
REG 2	.20	8.45	.5	.5
OCR 3	18 51	15.30	.5	.5
KMG 4	31.25	-34.[illegible]	.5	.5
CSG 5	-61.58	-28.30	.5	.5
9R 6	-.38	.02	1.	.5

	X Coord. (n mi)	Y Coord. (n mi)	Instrument Error (deg rms-n mi rms for DME)	Make Good radius (n mi)
FIX ID				
9L 7	.34	.25	1.	.5
9R and runway 8	-.38	-.48	.5	.5
9L and runway 9	.34	-.25	.5	.5
DME ID				
ATL 1	-8.05	-2.42	2.0	
REG 2	.20	8.45	2.0	
OCR 3	18.51	15.30	2.0	
ILS ID				
9R 1	-.38	.84	.5	.5
9L 2	.34	1.40	.5	.5

DATE: 8/8/72

Flight Schedule Array (ISKED - Packed Array)

No.	MODE	INFO1	INFO2	Pointers ALT	Pointers SPD	CP MODE	INFO3	INFO4	Length of Seg. (ISDST)(n mi)
1	1	3	12	31	0	3	9.0		9.0
2	1	3		31	0	5	3.0		8.2
3	3	220		31	0	1	8.0	103	16.4
4	3	270		5	7	2	2.0	600	27.7
5	10	180	1	3	0	2	2.0	60	0.0
6	3	119		3	0	1	7.0	271	0.0
7	6	2	269	2	0	1	1.0	338	0.0
8	6	2	269	1	8	10	7.0	.5	0.0
9	1	3	54	31	0	3	7.4		7.4
10	1	3	54	31	0	5	3.0		8.9
11	1	1	325	31	0	3	9.0		9.0
12	1	1	325	31	0	1	3.0	264	7.4
13	3	130		0	6	1	3.0	262	1.7
14	3	130		0	6	1	3.0	253	6.7
15	3	180		5	7	1	3.0	245	3.9
16	3	270		5	7	2	2.0	600	10.7
17	1	3	264	31	0	3	13.0		3.0
18	1	3	264	31	0	1	1.0	325	6.3
19	3	180		0	6	1	3.0	254	4.1

DATE: 8/8/72

Flight Schedule Array (ISKED - Packed Array)

No.	MODE	INFO1	INFO2	Pointers ALT	Pointers SPD	CP MODE	INFO3	INFO4	Length of Seg. (ISDST)(n mi)
20	3	180		5	7	1	3.0	248	3.8
21	3	270		5	7	2	2.0	600	5.6
22	3	110		0	6	1	3.0	247	13.0
23	3	180		5	7	1	3.0	238	3.8
24	3	270		5	7	2	2.0	600	18.1
25	1	1	117	31	0	3	8.6		8.6
26	1	1	117	31	0	1	3.0	178	7.7
27	3	320		31	0	1	1.0	78	15.3
28	3	270		5	7	2	2.0	600	31.2
29	10	0	2	4	0	2	2.0	60	0.0
30	3	59		4	0	1	6.0	267	0.0
31	6	1	267	2	0	1	1.0	336	0.0
32	6	1	267	6	8	10	6.0	.5	0.0
33	1	3	178	31	0	3	11.4		11.4
34	1	3	178	31	0	1	1.0	117	6.9
35	1	5	-18	31	0	3	9.2		9.2
36	1	5	-18	31	0	1	1.0	232	9.7
37	3	40		0	6	3	1.9		1.9
38	3	40		0	6	1	4.0	149	5.0

DATE: 8/8/72

Flight Schedule Array (ISKED - Packed Array)

No.	MODE	INFO1	INFO2	Pointers ALT	Pointers SPD	CP MODE	INFO3	INFO4	Length of Seg. (ISDST) (n mi)
39	3	360		5	7	1	2.0	251	5.6
40	3	270		5	7	2	2.0	600	15.2
41	1	1	232	31	0	3	8.7		8.7
42	1	1	232	31	0	1	5.0	18	9.9
43	3	8		0	6	1	1.0	251	3.8
44	3	360		5	7	1	2.0	253	5.6
45	3	270		5	7	2	2.0	600	12.5
46	3	60		0	6	1	4.	146	7.9
47	3	360		5	7	1	2.0	246	5.4
48	3	270		5	7	2	2.0	600	18.8
49	4		2.0	0	0	2	2.	120	0
50	4		2.0	0	0	2	2.	60	0
51	3			0	0	8			0
52	9	2		0	0	2	2.		0
53	3	89		1	1	6	1	0	0
54	3	89		7	6	1	3	213	3.8
55	3	89		7	5	1	3	200	5.[illegible]
56	3	10		31	4	1	8	100	5.2
57	3	320		31	5	1	3	220	4.2

DATE: 8/8/72

Flight Schedule Array (ISKED - Packed Array)

No.	MODE	INFO1	INFO2	Pointers ALT	Pointers SPD	CP MODE	INFO3	INFO4	Length of Seg. (ISDST) (n mi)
58	3	326		31	4	1	8	20	13.0
59	3	360		31	4	8	0	0	20.2
60	3	326		31	4	8	0	0	22.0
61	3	89		31	5	1	3	166	11.0
62	3	89		31	4	8	0	0	20.3
63	3	60		31	5	1	3	170	10.3
64	3	70		31	4	8	0	0	21.6
65	3	90		31	4	8	0	0	22.0
66	3	118		31	5	1	1	60	5.1
67	3	193		31	5	1	1	80	4.5
68	3	193		31	4	1	1	100	3.3
69	3	193		31	4	1	1	150	7.7
70	3	188		31	4	8	0	0	21.0
71	3	150		31	4	8	0	0	20.9
72	3	148		9	1	[illegible]	1	0	0
73	3	148		7	6	1	1	43	4.5
74	3	148		7	5	1	1	87	3.5
75	3	148		31	4	1	1	100	4.2
76	3	270		31	4	1	8	184	8.0

DATE: 8/8/72

Flight Schedule Array (ISKED - Packed Array)

No.	MODE	INFO1	INFO2	Pointers ALT	Pointers SPD	CP MODE	INFO3	INFO4	Length of Seg. (ISDST)(n mi)
77	3	270		31	4	1	8	203	6.3
78	3	270		31	4	1	8	237	19.8
79	3	270		31	4	8	0	0	7.2
80	3	320		31	4	8	0	0	13.1
81	9	2		31	0	2	2		0
82	3			31	0	8			0
83	3	270		31	5	2	2	60	7.2

DATE: 8/8/72

Controller Action Array (ICNTRA - Packed Array)

No.	Type	x North (n mi)	y East (n mi)	Checkpoint No. (ICHKPT)*	Controller option (IOPTN) or Airspace pointer if type 9 or 10 (ASPACE)	Next Controller Action (ICNTRA)	Controller No.	Distance to specified checkpoint from preceding checkpoint (n.mi.)
1	1	35.3	18.9	1	1	2	1	17.2
2	3	26.5	17.0	2	2	3	1	16.4
3	9	17.8	14.7	2	1	4	1	0
4	3	13.6	11.2	3	3	5	1	27.7
5	10	6.0	4.0	2	1	6	1	0
6	4	6.0	-9.0	4	4	0	5	0
7	1	28.0	28.4	1	9	8	1	16.3
8	3	23.2	21.8	2	2	3	1	16.4
9	1	28.4	-28.2	5	10	10	1	16.4
10	3	21.2	-23.1	6	5	11	1	23.0
11	4	9.6	-12.3	4	4	0	5	0
12	4	14.3	-17.9	4	4	0	5	0
13	1	-25.1	31.2	7	11	14	2	16.3
14	3	-21.3	23.6	8	12	15	2	15.3
15	9	-14.6	13.9	7	2	16	2	0
16	3	-12.2	11.9	9	13	17	2	31.2
17	10	-6.0	3.0	7	2	18	2	0
18	4	-6.0	-8.0	10	8	0	6	0

*Note: For type 9, use number of array following Airspace check.

DATE: 8/8/72

Controller Action Array (ICNTRA - Packed Array)

No.	Type	x North (n mi)	y East (n mi)	Checkpoint No. (ICHKPT)*	Controller option (IOPTN) or Airspace pointer if type 9 or 10 (ASPACE)	Next Controller Action (ICNTRA)	Controller No.	Distance to specified checkpoint from preceding checkpoint (n.mi.)
19	1	13.0	-37.9	5	14	20	1	19.3
20	3	14.0	-28.0	6	5	11	1	23.0
21	1	-36.1	17.3	7	15	22	2	18.3
22	3	-25.6	16.9	8	12	15	2	15.3
23	1	-34.8	-19.6	11	16	24	2	18.9
24	3	-24.6	-16.3	12	17	25	2	27.7
25	9	-16.2	-13.2	8	3	32	2	0
26	9	-16.2	-13.2	8	3	29	2	0
27	9	-16.2	-13.2	8	3	34	2	0
28	4	-7.5	-9.2	10	8	0	6	0
29	4	-10.2	-11.9	10	8	33	6	0
30	1	-28.2	-28.4	11	20	31	2	18.6
31	3	-21.8	-20.1	12	17	25	2	27.7
32	10	-7.8	-9.2	8	3	28	2	0
33	10	-7.4	-11.9	8	3	0	2	0
34	10	-7.9	-5.6	8	3	18	2	0
35								
36	2	.34	-.25	14	22	37	3	3.8

*Note: For type 9, use number of array following Airspace check.

DATE: 8/8/72

Controller Action Array (ICNTRA - Packed Array)

No.	Type	x North (n mi)	y East (n mi)	Checkpoint No. (ICHKPT)*	Controller option (IOPTN) or Airspace pointer if type 9 or 10 (ASPACE)	Next Controller Action (ICNTRA)	Controller No.	Distance to specified checkpoint from preceding checkpoint (n.mi.)
37	5	.4	3.0	15	51	38	7	5.2
38	3	.5	7.0	16	23	39	7	9.4
39	3	8.3	7.8	17	24	40	7	13.0
40	3	14.1	3.9	18	25	0	7	20.2
41	2	.34	-.25	14	27	42	3	3.8
42	5	.4	3.0	15	52	43	7	5.2
43	3	.5	7.0	20	28	44	7	11.0
44	3	.6	15.0	24	29	0	7	20.3
45	3	3.9	14.7	22	31	0	7	21.6
46	2	.34	-.25	14	33	47	3	3.8
47	3	.4	3.0	25	34	48	8	9.6
48	3	-3.9	7.6	26	35	49	8	3.3
49	3	-7.9	6.7	27	36	50	8	9.9
50	3	-13.9	5.3	28	37	0	8	21.0
51	2	.71	-.13	30	39	52	4	4.5
52	3	-2.0	1.5	31	40	53	8	3.5
53	3	-4.1	2.9	32	41	54	8	4.2
54	3	-6.9	4.6	27	42	50	8	9.9

*Note: For type 9, use number of array following Airspace check.

DATE: 8/8/72

Controller Action Array (ICNTRA - Packed Array)

No.	Type	x North (n mi)	y East (n mi)	Checkpoint No. (ICHKPT)*	Controller option (IOPTN) or Airspace pointer if type 9 or 10 (ASPACE)	Next Controller Action (ICNTRA)	Controller No.	Distance to specified checkpoint from preceding checkpoint (n.mi.)
55	2	.71	-.13	30	43	56	4	4.5
56	3	-2.0	1.5	31	44	57	8	3.5
57	3	-4.1	2.9	40	45	58	8	7.2
58	3	-7.5	4.0	33	46	59	8	8.6
59	3	-9.6	-3.0	34	47	60	8	6.3
60	3	-9.6	-7.0	35	48	61	8	19.8
61	3	-9.6	-20.0	36	49	0	8	7.2
62	6			1	53	63	1	
63	11			1	54	64	1	5.0
64	3	50	50	2	2	3	1	16.4
65	6			38	53	66	1	
66	9	50	50	2	1	67	1	0
67	11			38	54	68	1	5.0
68	3	50	50	3	3	5	1	27.7
69	6			5	53	70	1	
70	11			5	54	71	1	5.0
71	3	50	50	6	5	11	1	23.3
72	6			7	53	73	2	

*Note: For type 9, use number of array following Airspace check.

DATE: 8/8/72

Controller Action Array (ICNTRA - Packed Array)

No.	Type	x North (n mi)	y East (n mi)	Checkpoint No. (ICHKPT)*	Controller option (IOPTN) or Airspace pointer if type 9 or 10 (ASPACE)	Next Controller Action (ICNTRA)	Controller No.	Distance to specified checkpoint from preceding checkpoint (n.mi.)
73	11			7	54	74	2	5.0
74	3	50	50	8	12	15	2	15.3
75	6			39	53	76	2	
76	9	50	50	7	2	77	2	0
77	11			39	54	78	2	5.0
78	3	50	50	9	13	17	2	31.2
79	6			11	53	80	2	
80	11			11	54	81	2	5.0
81	3	50	50	12	17	25	2	27.7

*Note: For type 9, use number of array following Airspace check.

DATE: 8/8/72

Conflict Checkpoint Array (ICHKPT - Packed Array)

No.	No. of ending ETA Point	No. of Beginning ETA Point	Separation (n.mi.)	Maintain Separation Ahead (n.mi.)	Maintain Separation Behind (n.mi.)	Subscript for Airspace Conflict (ASPACE)
1	1	0	5	0	14.	0
2	2	1	3	0	0	0
3	3	2	3	0	16	0
4	4	3	3	0	5.8	0
5	5	0	5	0	14.	0
6	3	5	3	0	16	0
7	6	0	5	0	14.	0
8	7	6	3	0	0	0
9	8	7	3	0	18	0
10	9	8	3	0	5.8	0
11	10	0	5	0	14.	0
12	8	10	3	0	18	0
13						
14	12	0	3	3	3	0
15	13	12	3	0	0	0
16	14	13	3	5	5	1
17	15	14	3	0	5	0
18	16	15	3	0	5	0
19	17	15	3	0	5	0

DATE: 8/8/72

Conflict Checkpoint Array (ICHKPT - Packed Array)

No.	No. of ending ETA Point	No. of Beginning ETA Point	Separation (n.mi.)	Maintain Separation Ahead (n.mi.)	Maintain Separation Behind (n.mi.)	Subscript for Airspace Conflict (ASPACE)
20	18	13	3	5	5	0
21	19	13	3	5	5	0
22	20	19	3	0	5	0
23	21	19	3	0	5	0
24	21	18	3	0	5	0
25	23	12	3	0	0	2
26	24	23	3	0	3	0
27	25	24	3	0	3	0
28	26	25	3	0	5	0
29	27	25	3	0	5	0
30	28	0	3	3	3	0
31	29	28	3	0	0	2
32	24	29	3	0	0	0
33	30	36	3	3	3	3
34	31	30	3	3	0	3
35	32	31	3	3	3	0
36	33	32	3	0	3	0
37	34	32	3	0	3	0
38	2	0	3	0	0	0

Page 3 of 3

DATE: 8/8/72

Conflict Checkpoint Array (ICHKPT - Packed Array)

No.	No. of ending ETA Point	No. of Beginning ETA Point	Separation (n.mi.)	Maintain Separation Ahead (n.mi.)	Maintain Separation Behind (n.mi.)	Subscript for Airspace Conflict (ASPACE)
39	7	0	3	0	0	0
40	36	29	3	0	0	0

Page 1 of 1

Date 8/8/72

ETA Checkpoint Coordinates (IETCOOR - Packed Array)

ETA No.	X Coord. North (n mi)	Y Coord. East (n mi)		ETA No.	X Coord. North (n mi)	Y Coord. East (n mi)		ETA No.	X Coord. North (n mi)	Y Coord. East (n mi)
1	18.50	15.30		20	13.00	37.90				
2	6.00	4.70		21	5.60	39.60				
3	6.00	-23.00		22	1.20	40.00				
4	0	0		23	-6.40	7.00				
5	15.00	-18.70		24	-9.60	6.30				
6	-17.80	16.60		25	-19.20	4.00				
7	-6.00	6.80		26	-40.00	1.10				
8	-6.00	-24.40		27	-37.30	14.40				
9	0	0		28	-3.10	2.20				
10	-16.80	-13.70		29	-6.00	4.10				
11				30	-9.60	-5.60				
12	0.40	3.50		31	-9.60	-11.80				
13	0.50	8.70		32	-9.60	-31.60				
14	9.10	7.30		33	-9.60	-38.80				
15	19.80	0		34	0.40	-40.00				
16	40.0	0		35						
17	38.00	-12.40		36	-9.60	3.00				
18	0.70	19.70								
19	5.60	17.60								

DATE: 8/8/72

Controller Option Array (IOPTN - Packed Array)

No.	Pointer to Desired Altitude Array (ALTOPT)	Desired Speed Pointer (IPERFS)	Nominal Path Pointer (NNPATH)	Computed Vector Option (0=no, 1=yes)	Divergent Path Pointer (IDROUTE)	Altitude Option (0=no, 1=yes)	Speed Change Option (0=no, 1=yes)	Standard Hold Option (0=no, 1=yes)	Emergency Hold Option (0=no, 1=yes)
1	7	6	1	0	0	0	1	0	0
2	1	7	2	0	0	0	0	1	1
3	2	7	3	0	0	0	0	0	1
4	3	8	4	1	0	0	0	0	0
5	2	7	7	0	1	0	0	1	1
6	2	7	9	0	0	0	0	1	1
7	2	7	10	0	0	0	0	1	1
8	6	8	14	1	0	0	0	0	0
9	7	6	5	0	0	0	1	0	0
10	7	6	6	0	0	0	1	0	0
11	7	6	11	0	0	0	1	0	0
12	1	7	12	0	0	0	0	1	1
13	2	7	13	0	0	0	0	0	1
14	7	6	8	0	0	0	1	0	0
15	7	6	15	0	0	0	1	0	0
16	7	6	16	0	0	0	1	0	0
17	2	7	17	0	3	0	0	1	1
18	2	7	19	0	0	0	0	1	1
19	2	7	20	0	0	0	0	1	1

DATE: 8/8/72

Controller Option Array (IOPTN - Packed Array)

No.	Pointer to Desired Altitude Array (ALTOPT)	Desired Speed Pointer (IPERFS)	Nominal Path Pointer (NNPATH)	Computed Vector Option (0=no, 1=yes)	Divergent Path Pointer (IDROUTE)	Altitude Option (0=no, 1=yes)	Speed Change Option (0=no, 1=yes)	Standard Hold Option (0=no, 1=yes)	Emergency Hold Option (0=no, 1=yes)
20	7	6	18	0	0	0	1	0	0
21									
22	4	6	24	0	0	0	0	0	0
23	5	5	26	0	0	5	0	0	0
24	5	4	27	0	0	0	1	0	0
25	5	4	28	0	6	0	1	0	1
26	5	4	29	0	0	0	1	0	1
27	4	6	24	0	0	0	0	0	0
28	5	5	30	0	7	0	0	0	0
29	5	4	31	0	0	0	1	0	1
30	5	5	32	0	0	0	1	0	0
31	5	4	33	0	8	0	1	0	1
32	5	4	34	0	0	0	1	0	1
33	4	6	24	0	0	0	0	0	0
34	5	5	35	0	0	5	0	0	0
35	5	4	36	0	0	5	0	0	0
36	5	4	37	0	0	5	1	0	0
37	5	4	38	0	9	0	1	0	1
38	5	4	39	0	0	0	1	0	1

DATE: 8/8/72

Controller Option Array (IOPTN - Packed Array)

No.	Pointer to Desired Altitude Array (ALTOPT)	Desired Speed Pointer (IPERFS)	Nominal Path Pointer (NNPATH)	Computed Vector Option (0=no, 1=yes)	Divergent Path Pointer (IDROUTE)	Altitude Option (0=no, 1=yes)	Speed Change Option (0=no, 1=yes)	Standard Hold Option (0=no, 1=yes)	Emergency Hold Option (0=no, 1=yes)
39	4	6	40	0	0	0	0	0	0
40	4	5	41	0	0	4	0	0	0
41	5	4	42	0	0	5	0	0	0
42	5	4	37	0	0	5	0	0	0
43	4	6	40	0	0	0	0	0	0
44	4	5	41	0	0	4	0	0	0
45	5	5	49	0	0	5	0	0	0
46	5	4	43	0	0	5	1	0	0
47	5	4	44	0	0	5	0	0	0
48	5	4	45	0	0	5	1	0	0
49	5	4	46	0	10	0	0	0	0
50	5	4	47	0	0	0	0	0	0
51	4	5	25	0	0	0	0	0	0
52	4	5	25	0	0	0	0	0	1
53	1	6	48	0	0	0	1	0	1
54	1	6	22	0	0	0	0	0	0

Page 1 of 1

DATE: 8/8/72

Altitude Option Array (ALTOPT)

Option No. (I)	Pointer From Option Array (J)								
	1	2	3	4	5	6	7		
1	6000 / 10000	6000 / 6000	1024 / 1024	4000 / 4000	15000 / 35000	1015 / 1015	6000 / 11000		
2	0 / 11000	0 / 0	0 / 0	0 / 0	5000 / 4000	0 / 0	0 / 0		
3	0	0	0	0	0	0	0 / 0		

K = 1: prop

K = 2: jet

prop	
	jet

Page 1 of 1

DATE: 8/8/72

Divergent Path Array (IDROUTE - Packed Array)

No.	Pointer to Checkpoint Array (ICHKPT)	Pointer to Option Array (IOPTN)	Pointer to Controller Action Array (ICNTRA)	Pointer to Another Divergent Route (IDROUTE)	Distance to specified checkpoint from preceding checkpoint (n.mi.)
1	6	6	12	2	14.9
2	6	7	6	0	35.6
3	12	18	26	4	23.6
4	12	19	31	0	34.1
5					
6	19	26	0	0	22.0
7	21	30	45	0	10.2
8	23	32	0	0	22.0
9	29	38	0	0	20.9
10	37	50	0	0	13.6

Page 1 of 4

DATE: 3/14/73

Path Number Array (NNPATH) [illegible]

No.	No. of segments	Pointer to Next Optional Vector	Path No.	Subscripts to Schedule Array (ISKED)			
				1st Segment	2nd Segment	3rd Segment	4th Segment
1	2	0	1	1	2	0	0
2	1	0	2	3	0	0	0
3	1	0	3	4	0	0	0
4	4	0	4	5	6	7	8
5	2	0	5	9	10	0	0
6	2	0	6	11	12	0	0
7	4	0	7	13	14	15	16
8	2	0	8	17	18	0	0
9	4	0	9	13	19	20	21
10	4	0	10	13	22	23	24
11	2	0	11	25	26	0	0
12	1	0	12	27	0	0	0
13	1	0	13	28	0	0	0
14	4	0	14	29	30	31	32
15	2	0	15	33	34	0	0
16	2	0	16	35	36	0	0
17	4	0	17	37	38	39	40
18	2	0	18	41	42	0	0
19	4	0	19	37	43	44	45

DATE: 3/14/72

Path Number Array (NNPATH) Arrivals

No.	No. of-segments	Pointer to Next Optional Vector	Path No.	Subscripts to Schedule Array (ISKED)			
				1st Segment	2nd Segment	3rd Segment	4th Segment
20	4	0	20	37	46	47	48
21	2	0	21	49	50	0	0
22	1	0	22	51	0	0	0
23	1	0	23	52	0	0	0

DATE: 3/16/72

Path Number Array (NNPATH) departures

No.	No. of-segments	Pointer to Next Optional Vector	Path No.	Subscripts to Schedule Array (ISKED)			
				1st Segment	2nd Segment	3rd Segment	4th Segment
24	2	0	24	53	54	0	0
25	1	0	25	55	0	0	0
26	2	0	26	56	57	0	0
27	1	0	27	58	0	0	0
28	1	0	28	59	0	0	0
29	1	0	29	60	0	0	0
30	1	0	30	61	0	0	0
31	1	0	31	62	0	0	0
32	1	0	32	63	0	0	0
33	1	0	33	64	0	0	0
34	1	0	34	65	0	0	0
35	2	0	35	66	67	0	0
36	1	0	36	68	0	0	0
37	1	0	37	69	0	0	0
38	1	0	38	70	0	0	0
39	1	0	39	71	0	0	0
40	2	0	40	72	73	0	0
41	1	0	41	74	0	0	0
42	1	0	~~4~~42	75	0	0	0

Page 4 of 4

DATE: 3/16/72

Path Number Array (NNPATH) departures

No.	No. of-segments	Pointer to Next Optional Vector	Path No.	Subscripts to Schedule Array (ISKED) 1st Segment	2nd Segment	3rd Segment	4th Segment
43	1	0	43	76	0	0	0
44	1	0	44	77	0	0	0
45	1	0	45	78	0	0	0
46	1	0	46	79	0	0	0
47	1	0	47	80	0	0	0
48	2	0	48	81	82	0	0
49	1	0	49	83	0	0	0

Date 8/8/72

Limit A/C Array (LIMAC - Packed Array)

Sub-script from IOPTN	ETA Array (1) or Waiting Queue (2) Indicator	Subscript of Number in ETA array (NETA) or Number in Queue (IQUEUE)	Upper Limit of No. of A/C Before Additional Separation Applied	Lower Limit of No. of A/C Before Additional Separation Removed	Separation to be Added	Pointer to Switch Indicating Whether Additional Separation was Previously Applied (SWITCH)	Pointer to Next Possible Limit A/C Array (LIMAC)
1	1	3	8	7	0.0	1	-2
2	1	8	4	4	5.0	1	0
3	1	8	8	7	0.0	2	-4
4	1	3	4	4	5.0	2	0

Controller Message Length Array in Number of Scans
for Ith Type of Message
MSG(I) Array

I	Type of Message	Length of Time (in 4 sec scans) to Transmit
1	Arrival clearance to first check point	1.
2	Hold at check point 1	1.
3	Vectoring	1.
4	Speed control	1.
5	Normal flight to next check point (pseudo message)	1.
6	Sequencing	1.
7	Departure clearance	3.
8	Change altitude	1.
9	360° emergency hold	1.

Initial Controller Action
Subscripts for Each Route
(ISTARTA(I))

Route No. I	Controller Action Subscript (ICNTRA)
1	36
2	41
3	51
4	55
5	1
6	7
7	9
8	17
9	13
10	21
11	23
12	30

Scheduled Altitudes
(ALTSKD(I))

Altitude Type I	Altitude (ft)
1	1024
2	2500
3	3500
4	4500
5	6000
6	1015
7	4000
8	25000
9	986

Variation on Desired Speed
(SPDSTD(I,J))

Type Speed I	Prop Variation J = 1 (kts)	Jet Variation J = 2 (kts)
0	0	0

Aircraft Type Dependent Random Heading Errors
IACFPER(I) (packed array)

Type A/C I	Aircraft Error (deg rms)	Type A/C I	Aircraft Error (deg rms)
1	.5	12	1.0
2	.5	13	1.0
3	.5	14	1.0
4	.5	15	.5
5	.5	16	.5
6	.5	17	1.0
7	2.0	18	1.0
8	1.0	19	.5
9	1.0	20	.5
10	2.0	21	.5
11	.5		

Table B-1. Definition of miscellaneous input parameters and numerical values used for Atlanta simulation.

Variable Name	Value	Definition
AK	.9	constant that limits change of path error
ALPHK	1	pilot gain constant for VOR or ILS mode
ARRMNA	6000.	minimum arrival altitude in terminal area
ARRMXA	11000.	maximum arrival altitude in terminal area
BETA	75.	angle of view in degrees desired for the cockpit display of the runway
BOUND	40.1	radius in n mi which defines the circular boundary around the airspace being considered by the model
CGLIDE	.0125	scale factor for DAC output of the error off the glide slope for the pseudo-cockpit
CILS	.05	scale factor for DAC output of the error off the ILS for the pseudo-cockpit
CKRC	5.81	scale factor for the ADC input of the rate of climb for the pseudo-cockpit
CKROLL	-750.	scale factor for the ADC input of the roll angle for the pseudo-cockpit
CRC	.04	scale factor for DAC output of the rate of climb for the pseudo-cockpit
CROLL	.02	scale factor for the DAC output of the roll angle for the pseudo-cockpit
DELALT	50	"make good" limit on altitude in ft
DELSPD	9	"make good" limit on speed in kts
DELT	4	time increment in sec
DKCON	45	correction constant when flying DME mode
EHOLDD	120.	time in sec required for completion of one emergency hold
ENROTA(1)	6500.	altitude above which prop-type aircraft use enroute descent rate
ENROTA(2)	11500.	altitude above which jet uses its enroute descent rate
GLIDE(1)	2.88	glide slope angle for runway 9L
GLIDE(2)	2.60	glide slope angle for runway 9R
HDCON	.0025	scale factor for the DAC output of the heading for the pseudo-cockpit
IARG	7	small odd integer used as a seed for the random number generator associated with the radar errors

Table B-1. Continued.

Variable Name	Value	Definition
IARRN	4	subscript of ETA array for touchdown times for arrivals from the north
IARRS	9	subscript of ETA array for touchdown times for arrivals from the south
IDEPN	11	subscript of ETA array for take-off times for departures to the north
IDEPS	35	subscript of ETA array for take-off times for departures to the south
IHPATHE	23	pointer to NNPATH for emergency hold path description
IHPATHS	21	pointer to NNPATH for standard hold path description
IPOPPA	22	pointer to NNPATH for initial popup path description
ITANG(1,1)	109	angle (heading) for intersecting the final leg from base leg when distance from north ILS gate to intersection is $\leq$ 2 n mi
ITANG(1,2)	119	angle (heading) for intersecting the final leg from base leg when distance from north ILS gate to intersection is > 2 n mi
ITANG(2,1)	69	angle (heading) for intersecting the final leg from base leg when distance from south ILS gate to intersection is $\leq$ 2 n mi
ITANG(2,2)	59	angle (heading) for intersecting the final leg from base leg when distance from south ILS gate to intersection is > 2 n mi
J	36	temporary variable for number ETA point coordinates
JJ	7	temporary variable indicating the number of altitude regions
NADF	2	number of ADFs
NALTOP	3	maximum number of altitude options available to a controller in a particular region
NALTSKD	9	number of fixed altitudes to be used by the flight schedule
NARR	8	number of arrival routes
NCACT	81	number of controller action points
NCHK	40	number of checkpoints
NCP	6	number of estimated times of arrival at checkpoints to be output for an aircraft
NDEPR	4	number of departure routes
NDME	3	number of DMEs

Table B-1. Continued.

Variable Name	Value	Definition
NDRTE	10	number of divergent routes
NFIX	9	number of FIXs
NILS	2	number of ILSs
NLIM	4	number of air traffic flow control descriptors
NOPT	54	number of option descriptors
NPTH	49	number of legs that describe this route structure
NSKED	83	number of segments that describe this route structure
NSPDSTD	9	number of speeds which can be varied for conflict resolution
NSTART	12	number of initial controller action points
NTRAF	101	total number of aircraft in traffic sample
NTYPE	21	number of performance types of aircraft
NUMSC	2500	number of scans desired for run
NVOR	8	number of VORs
PASEP	40	time in sec added to gate time of next arrival on "other" runway which reserves the runway for the arrival and prohibits a departure
PDSEP	60	time separation in sec between departure on "other" runway and the departure requiring clearance
PHIK	1	constant to convert desired heading correction to bank angle
PPMX(1)	-4.	aircraft past position markers; pairs of x, y coordinates which are rotated and translated to be behind aircraft symbols on CRT
(2)	0	
(3)	-4.	
PPMY(1)	-4.	
(2)	0	
(3)	4.	
RAD(1)	0.	north-south coordinate for the location of the radar
RAD(2)	0.	east-west coordinate for the location of the radar
RAD(3)	1024.	altitude above sea level for the location of the radar
RCON	.5	radius in n mi around VOR within which "away from VOR" equations cannot be used
RCP(1)	14	maximum length in n mi for downwind leg of the approach pattern for arrivals from the north
RCP(2)	16.4	maximum length in n mi for downwind leg of the approach pattern for arrivals from the south

Table B-1. Continued.

Variable Name	Value	Definition
RWYX(1,1)	.365	north-south coordinates in n mi for corners of runway 9L which are used for the cockpit display of the runway
(2,1)	.315	
(3,1)	.315	
(4,1)	.365	
RWYX(1,2)	-.355	north-south coordinates in n mi for corners of runway 9R which are used for the cockpit display of the runway
(2,2)	-.405	
(3,2)	-.405	
(4,2)	-.355	
RWYY(1,1)	1.4	east-west coordinates in n mi for corners of runway 9L which are used for the cockpit display of the runway
(2,1)	1.4	
(3,1)	-.25	
(4,1)	-.25	
RWYY(1,2)	.84	east-west coordinates in n mi for corners of runway 9R which are used for the cockpit display of the runway
(2,2)	.84	
(3,2)	-.48	
(4,2)	-.48	
RWYZ(1)	1024.	altitude in ft of runway 9L
RWYZ(2)	1015.	altitude in ft of runway 9R which are used for the cockpit display of the runway
SASEP	40	time in sec added to gate time of next arrival on "same" runway which reserves the runway for the arrival and prohibits a departure
SEPP(1)	3	required separation in n mi between aircraft arriving on the same runway
SEPP(2)	2.911	required separation in n mi between aircraft arriving on different runways
SHOLDD	240	time in sec required for completion of one standard hold pattern
SIG(1)	380.	standard deviation in ft for range from the radar used to simulate radar errors
SIG(2)	.25	standard deviation in deg for bearing from the radar used to simulate radar errors
SIG(3)	81.	standard deviation in ft for altitude above the radar used to simulate radar errors
SQX(1)	0	six sets of x, y coordinates needed to draw symbol on CRT for pop-up aircraft
(2)	4	
(3)	0	
(4)	0	
(5)	-4	
(6)	0	
SQY(1)	4	
(2)	0	
(3)	-4	
(4)	4	
(5)	0	
(6)	-4	

Table B-1. Concluded.

Variable Name	Value	Definition
TRIX(1)	0	five sets of x, y coordinates needed to draw aircraft symbols on CRT
(2)	-4	
(3)	8	
(4)	-4	
(5)	0	
TRIY(1)	0	
(2)	4	
(3)	0	
(4)	-4	
(5)	0	
VECERR	3.	amount of heading error permissible without recalculating vector heading
VECSD	2	standard deviation for vectors
XA(1)	6.0	north-south coordinate of sequencing point on northern downwind segment
XA(2)	-6.0	north-south coordinate of sequencing point on southern downwind segment
XC(1)	6.0	north-south coordinate of final point for turnoff from northern downwind segment
XC(2)	-6.0	north-south coordinate of final point for turnoff from southern downwind segment
XDILS(1)	5.20	distance between the ILS gate and the runway 9L touchdown point
XDILS(2)	4.97	distance between the ILS gate and the runway 9R touchdown point
XDIST(1)	3.55	east-west distance in n mi between ILS gate and sequencing fix for arrivals from the north
XDIST(2)	2.55	east-west distance in n mi between ILS gate and sequencing fix for arrivals from the south
XFINAL(1)	.79	north-south coordinates of base leg intersection with 2° north of the ILS for runway 9L
XFINAL(2)	-1.67	north-south coordinates of base leg intersection with 2° south of the ILS for runway 9R
YA(1)	-9.0	east-west coordinate of sequencing point on northern downwind segment
YA(2)	-9.0	east-west coordinate of sequencing point on southern downwind segment
YC(1)	-23.0	east-west coordinate of final point for turnoff from northern downwind segment
YC(2)	-24.4	east-west coordinate of final point for turnoff from southern downwind segment
ZCON	.0001	scale factor for the DAC output of the altitude for the pseudo-cockpit

APPENDIX C

Input Data for the Traffic Sample Generation Program

This appendix contains data sheets used to define inputs to the traffic sample generation program. The data sheets are filled out with information currently being used to generate traffic samples with varying hourly operation rates.

A master traffic sample data sheet is also shown on page which is used if it is desired to use scheduled traffic for the traffic sample. Data on this sheet is output from the traffic sample generation program.

Page ___ of ____ Date ______________

TRAFFIC GENERATOR--AIRCRAFT INITIAL CONDITIONS

A/C No. I	A/C Type (ITYPE(I))	Initial Speed (Kts) (VO(I))	Speed Deviation (Kts, rms) (SV(I))	Altitude Deviation (ft rms) (SZ(I))	Lateral Deviation (n mi) (SPS(I))	Heading Deviation (deg rms) (HDG(I))	Max. Lat. Deviation (n mi) (MAXLAT(I))
1	2	250.	5.	100.	1.	5.	2.
2	2	250.	5.	100.	1.	5.	2.
3	2	250.	5.	100.	1.	5.	2.
4	2	250.	5.	100.	1.	5.	2.
5	2	250.	5.	100.	1.	5.	2.
6	2	250.	5.	100.	1.	5.	2.
7	1	140.	4.	100.	1.	5.	2.
8	1	165.	4.	100.	1.	5.	2.
9	1	200.	5.	100.	1.	5.	2.
10	1	140.	4.	100.	1.	5.	2.
11	2	250.	5.	100.	1.	5.	2.
12	1	165.	4.	100.	1.	5.	2.
13	1	200.	5.	100.	1.	5.	2.
14	1	200.	5.	100.	1.	5.	2.
15	2	250.	5.	100.	1.	5.	2.
16	2	250.	5.	100.	1.	5.	2.
17	1	200.	5.	100.	1.	5.	2.
18	2	250.	5.	100.	1.	5.	2.
19	2	250.	5.	100.	1.	5.	2.
20	1	250.	5.	100.	1.	5.	2.

Date ____________

TRAFFIC GENERATION--ROUTE STARTING CONDITIONS

No. J	Initial Point X North (n mi) (XO(J))	Initial Point Y East (n mi) (YO(J))	Initial Heading (degrees) (PSI(J))	N or S (1=N,2=S) (NORS(J))	Arr., Dep., or Pop-Up (1=Dep,2=Arr, 3=Pop-Up) (KAD(J))	Nominal Alt. (ft) (Z(J)) Prop	Nominal Alt. (ft) (Z(J)) Jet	No. of 1000 ft levels up (LUP(J,L)) prop L=1	No. of 1000 ft levels up (LUP(J,L)) jet L=2	No. of 1000 ft levels down (LDN(J,L)) prop L=1	No. of 1000 ft levels down (LDN(J,L)) jet L=2
1	0.24	-0.25	89.0	1	1	1000.	2500.	2	3	5	5
2	0.24	-0.25	87.0	1	1	"	"	2	3	5	5
3	0.71	-0.13	148.0	2	1	"	"	"	"	"	"
4	0.71	-0.13	[illegible]	2	1	"	"	"	"	"	"
5	35.30	18.90	142.0	1	2	"	"	"	"	"	"
6	28.10	28.50	234.0	1	2	"	"	"	"	"	"
7	28.40	-28.20	145.0	1	2	"	"	"	"	"	"
8	13.00	-31.90	84.0	1	2	"	"	"	"	"	"
9	-25.10	31.20	297.0	2	2	"	"	"	"	"	"
10	-36.10	17.30	358.0	2	2	"	"	"	"	"	"
11	-34.80	-17.60	15.0	2	2	"	"	"	"	"	"
12	-26.20	-28.40	52.0	2	2	"	"	"	"	"	"

Page 1 of 1 Date ______

TRAFFIC GENERATION--MISC. CONSTANTS AND ROUTE, TYPE PROBABILITIES

No. of Routes (JM)	12
No. of Types (IM)	20
Total Sample Time (min) (KM)	300
Average Operations per Hour (KOP)	40
Maximum Allowed Time Between Operations (secs) (KTM)	270
Random Number Constant (small odd integer) (KARG)	13
Radius of Arrival Circle (n mi) (RAD)	40.0

Route No. J	Probability of Occurrence	Cumulative Probability (B(J))
1	.05	0.050000
2	.10	0.150000
3	.05	0.200000
4	.10	0.300000
5	.093902	0.393902
6	.093903	0.487805
7	.085366	0.573171
8	.085366	0.658537
9	.085365	0.743902
10	.085366	0.829268
11	.085366	0.914634
12	.085366	1.000000
13		
14		

Type No. I	Probability of Occurrence	Cumulative Probability (C(I))
1	.0048	0.0048
2	.0917	0.0965
3	.1110	0.2075
4	.3480	0.5555
5	.0484	0.6039
6	.0049	0.6088
7	.0290	0.6378
8	.0049	0.6427
9	.0532	0.6959
10	.0097	0.7056
11	.0097	0.7153
12	.0048	0.7201
13	.0048	0.7247
14	.0580	0.7829
15	.0917	0.8746
16	.0194	0.8940
17	.0145	0.9085
18	.0194	0.9279
19	.0580	0.9859
20	.0141	1.0

Page / of /

MASTER TRAFFIC SAMPLE

Date

(ITRAF1 - Packed Array)				(ITRAF2 - Packed Array)								
A/C ID	Initial Coordinates x North (n mi)	y East (n mi)	z Alt. (ft)	Initial Speed (kts.)	Initial Heading (deg.)	Perfor-mance Class	Jet(1) or Prop(2)	Route No.	Offer Time (sec.)	Dep.(1) or Arr.(2)	North(1) or South(2)	Desired Altitude (ft.)

APPENDIX D

Subroutine Descriptions

This appendix contains a brief description of the function of each of the subroutines developed for the program. In addition to the subroutines described herein, numerous system-supplied subroutines are used. These subroutines are defined in documentation of the Langley Research Center computer complex.

SUBROUTINE DESCRIPTIONS

OVERLAY	NAME	FUNCTION
0,0	TAATM	Overlay 0,0 that sets up common resident core storage needed by all other overlays--calls a system routine that loads in the next overlay.
	PAKSPL (Special Pack)	Packs one or several variables into one computer core word based on a table of the variables' attributes.
	UPKSPL (Special Unpack)	Unpacks from one computer core word one or several variables based on a table of the variables' attributes.
	ISHFTE	COMPASS routine that performs right shift with sign extension.
	PAKUNI (Uniform Pack)	Packs one or more variables, of uniform attributes, into one or more contiguous computer core words.
	UPKUNI (Uniform Unpack)	Unpacks one variable from a linear array of contiguous computer core words which contains several variables with uniform attributes.
	LSHFT	COMPASS routine that performs left circular shift.
1,0	TATINT	Reads in data in non-real-time.
	INIT	Initializes variables and sets constants.
2,0	TATPRNT	Writes various arrays, e.g. active traffic, on tape and on printer.
3,0	TATREAD	Reads in additional traffic data.
4,0	TATMAIN	Jumping off and return to point. Main program of overlay Number 4.
4,1	TATREST	Reset Section - Performs initialization of simulation and reset after simulation has been started.
4,2	TATHOLD	Hold Section - Develops CRT static display of routes, range markers (rings), runways, and controller message headers.
	MAKTAPE	Rewinds the CRT file and copies to a tape for later batch processing with a program to plot the CRT display on the CALCOMP plotter.
	DASHCIR	Plots a circular range marker of a desired radius on the CRT screen.

OVERLAY	NAME	FUNCTION
4,3	TATOPER	Operate Section - Updates scan and frame counters. Calls routines for tracking, communications, controller actions, and CRT displays.
	TRACKAC	Track Aircraft - Solves aircraft dynamics - introduces path errors - deletes aircraft which have left the terminal area or which have landed - checks whether controller action point has been passed. Updates active schedule upon meeting objectives for a particular schedule segment, e.g. has "made good" a range from a Nav. site objective. Updates aircraft position.
	RANDU	Computes uniformly distributed random real numbers between 0 and 1.0.
	COMMUN	Communication Channels Simulation - Adds aircraft to controller action arrays from departure and arrival queues. Delivers messages. Modifies schedules according to message contents. Adds clearances for conflict-free segments.
	CNTRLAC	Control Aircraft - Deals with specific controller actions needed, e.g. arrival, departure, hand-off, airspace conflict, general conflict, sequencing actions. Uses options available for the particular controller action point to resolve traffic conflicts.
	ETACOMP	Estimated Time of Arrival Computation - ETA at a point or, based on a switch, the ETA at a separation distance from the point. Takes into consideration altitude and speed changes. Also returns earliest and latest ETAs based on when speed change occurs.
	CONCHK	Conflict Check. Does the actual checking for any conflicts based on arrival times calculated for a particular option. Returns binary switch set for conflict/no conflict.
	SPDCNTR	Speed Control - Computes whether a change of speed, and when to apply the change, will yield a conflict-free ETA envelope at a checkpoint, i.e, a slot in the ETA. Passing another aircraft may be allowed under some conditions.
	COMETA	Compute Estimated Time of Arrival based on speed and distance.
	CRTDSPL	CRT Display - Drives display routines to display for each aircraft in active traffic an aircraft symbol, legend box associated with the aircraft, two past position markers, hand-off or hold information, and controller messages. Static display of routes, etc., is background information.

OVERLAY	NAME	FUNCTION
4,3	XFORM	Performs relative coordinate transformation.

APPENDIX E

Definitions of Variables for the Terminal Area Simulation Model

This appendix contains a complete list of variables used in the model. For each variable, a definition, its dimension(s), aliases assigned for programming convenience by FORTRAN-equivalence statements, and a list of routines in which each variable is used are included when applicable.

Table E-1. Definitions of Variables.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
A			CNTRLAC	distance between time of turnoff of downwind leg to time of initiating turn onto localizer course
AALT	RESTOR(4)			see RESTOR
ACCEL			COMETA	acceleration rate for particular aircraft under consideration
ACCRT			TRACKAC	computed acceleration rate needed by aircraft to obtain a desired speed on the next scan
ACERR			TRACKAC	aircraft type-dependent flight path error for particular aircraft under consideration
ACLS(I)		(50)	CRTDSPL,INIT	array of machine instructions for displaying on the CRT the alphanumerics for the type of aircraft
ADC(I)		(32)	TATINT,TATREST,TATOPER	variable referencing the analog-to-digital converters
AK	TABLE(1)			see TABLE
ALPHA			TATOPER,TRACKAC	temporary used for an angle
ALPHK	TABLE(2)			see TABLE
ALT			TRACKAC	present altitude for aircraft under consideration
ALTF			TRACKAC	present altitude goal for aircraft under consideration
ALTLIM			CNTRLAC	altitude limit above which the departure under consideration cannot fly
ALTOPT(I,J,K)		(3,6,2)	CNTRLAC,TATINT	see input data
ALTSKD(I)		(9)	CNTRLAC,TATINT	see input data

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
ANG(I)	SEQUE(1,14) SEQUE(3,14)			see SEQUE
ANGLE			CNTRLAC	temporary used for the angle of intersection of the base leg with the final leg of the sequencing pattern for an aircraft under consideration
ANG2			CNTRLAC	temporary
ARDIF			TRACKAC	degrees off a radial that an aircraft is trying to intersect
ARRMNA			CNTRLAC,TATINT	see input data
ARRMXA			CNTRLAC,TATINT	see input data
AR1X			TATHOLD	part of route definition for background display of terminal area on CRT
AR1Y			TATHOLD	same as AR1X
AR2X			TATHOLD	same as AR1X
AR2Y			TATHOLD	same as AR1X
AR3X			TATHOLD	same as AR1X
AR3Y			TATHOLD	same as AR1X
AR4X			TATHOLD	same as AR1X
AR4Y			TATHOLD	same as AR1X
AR5X			TATHOLD	same as AR1X
AR5Y			TATHOLD	same as AR1X
AR6X			TATHOLD	same as AR1X
AR6Y			TATHOLD	same as AR1X

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
AR7X			TATHOLD	same as AR1X
AR7Y			TATHOLD	same as AR1X
AR8X			TATHOLD	same as AR1X
AR8Y			TATHOLD	same as AR1X
ASKSPD(I,J)		(35,11)	CNTRLAC,COMMUN,TRACKAC	scheduled speeds for each segment assigned each aircraft in active traffic
ASPACE1(I,J)		(4,5)	CNTRLAC	J aircraft ID numbers in a particular I airspace and under the control of a particular controller; another controller can use this airspace only if the airspace is empty at the altitude level he desires to use
ASPACE2(I,J)		(4,5)	CNTRLAC	altitudes at beginning of path for J aircraft in a particular I airspace and under the control of a particular controller; another controller can use this airspace only if the airspace is empty at the altitude level he desires to use
ASPACE3(I,J)		(4,5)	CNTRLAC	altitudes at the terminal end of a path for J aircraft in a particular I airspace and under the control of a particular controller; another controller can use this airspace only if the airspace is empty at the altitude level he desires to use
ATRAF(I,J)	IATRAF(I,J)	(35,26)	CNTRLAC,CRTDSPL,TATREST, TRACKAC	tracking information for each I aircraft in active traffic
for I^{th} aircraft J = 1	IATRAF1(I) IATRAF(I,1)	(35)	CNTRLAC,COMMUN,CRTDSPL, TRACKAC	aircraft ID number
2	ATRAF2(I)	(35)	CNTRLAC,CRTDSPL,TRACKAC	x - north-south coordinate (n mi)
3	ATRAF3(I)	(35)	CNTRLAC,CRTDSPL,TRACKAC	y - east-west coordinate (n mi)

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
4	ATRAF4(I)	(35)	CNTRLAC,COMMUN,CRTDSPL, TRACKAC	z - altitude (ft)
5	ATRAF5(I)	(35)	CNTRLAC,COMMUN,CRTDSPL, TRACKAC	speed (n mi/sec)
6	ATRAF6(I)	(35)	CNTRLAC,CRTDSPL,TRACKAC	heading (deg)
7	IATRAF7(I) IATRAF(I,7)	(35)	CNTRLAC,CRTDSPL,TRACKAC	performance class
8	IATRAF8(I) IATRAF(I,8)	(35)	CNTRLAC,TRACKAC	prop(1) or jet(2)
9	IATRAF9(I) IATRAF(I,9)	(35)	CNTRLAC,TRACKAC	route number
10	IATRF10(I) IATRAF(I,10)	(35)	TRACKAC	scheduled time of entry in scans
11	IATRF11(I) IATRAF(I,11)	(35)	CNTRLAC,TRACKAC	departure(1) or arrival(2)
12	IATRF12(I) IATRAF(I,12)	(35)	CNTRLAC,TRACKAC	north(1) or south(2) region
13				VX - velocity in x (n mi/sec)
14				VY - velocity in y (n mi/sec)
15				VZ - velocity in z (ft/sec)
16	IATRF16(I) IATRAF(I,16)	(35)	CNTRLAC,CRTDSPL,TRACKAC	current flight mode times 10 plus control point mode
17	IATRF17(I) IATRAF(I,17)	(35)	CNTRLAC,TRACKAC	current flight leg
18	ATRF18(I)	(35)	CNTRLAC,TRACKAC	radar x (n mi)
19	ATRF19(I)	(35)	CNTRLAC,TRACKAC	radar y (n mi)
20	ATRF20(I)	(35)	CNTRLAC,TRACKAC	radar z (ft)

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
21	IATRF21(I) IATRAF(I,21)	(35)	CNTRLAC,COMMUN,TRACKAC	current path; packed word
22	IATRF22(I) IATRAF(I,22)	(35)	CNTRLAC,COMMUN,TRACKAC	current flight clearances remaining
23	IATRF23(I) IATRAF(I,23)	(35)	CNTRLAC,COMMUN,TRACKAC	future path; packed word
24	IATRF24(I) IATRAF(I,24)	(35)	CNTRLAC,COMMUN,TRACKAC	future path clearances received
25	IATRF25(I) IATRAF(I,25)	(35)	TRACKAC	next controller action subscript (ICNTRA)
26	ATRF26(I)	(35)	CNTRLAC	altitude limit for departures (ft)
ATRAF2(I)	ATRAF(I,2)			see ATRAF
ATRAF3(I)	ATRAF(I,3)			see ATRAF
ATRAF4(I)	ATRAF(I,4)			see ATRAF
ATRAF5(I)	ATRAF(I,5)			see ATRAF
ATRAF6(I)	ATRAF(I,6)			see ATRAF
ATRF18(I)	ATRAF(I,18)			see ATRAF
ATRF19(I)	ATRAF(I,19)			see ATRAF
ATRF20(I)	ATRAF(I,20)			see ATRAF
ATRF26(I)	ATRAF(I,26)			see ATRAF
A1X, A1Y to A8X, A8Y			TATHOLD	part of route definition for background display of terminal area on CRT
B			CNTRLAC	distance flown from time of turn-off base leg to intersection with localizer course plus distance on localizer course from a line that is the perpendicular bisector of the localizer course from the earliest turn-off point to the ILS gate

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
BETA	TABLE(26)			see TABLE
BOUND			TATINT,TRACKAC	see input data
BUFF(I)		(17)	MAKTAPE	storage buffer for copying CRT file to tape
C			TATHOLD	scaling factor for CRT used to scale n mi to CRT units
CETA			CNTRLAC,ETACOMP,TATREST	difference between actual time of arrival and estimated time of arrival at a particular previously-passed ETA point; used as a correction factor for estimating the time of arrival at a new ETA point
CGLIDE			TATINT,TATOPER	see input data
CHEAD			TRACKAC	temporary for the new corrected heading for a given aircraft on a given scan
CILS			TATINT,TATOPER	see input data
CKRC	TABLE(54)			see TABLE
CKROLL	TABLE(53)			see TABLE
CO			XFORM	used when displaying the terminal area relative to a given aircraft and defined as the cosine of the heading of this aircraft
COAR			CRTDSPL,XFORM	used when displaying the terminal area relative to the location of a given aircraft and defined as the cosine of the angle between the i^{th} aircraft and the aircraft to which the terminal area is relative
CONFLCT			CNTRLAC,CONCHK,SPDCNTR, TATREST	yes(1) or no(2) indicator as to whether or not the particular aircraft under consideration has a conflict that needs to be resolved with another aircraft

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
CPCOORD(I)		(2)	TRACKAC	temporary name for coordinates of a particular controller action point under consideration
I = 1	XCP		TRACKAC	north-south coordinate (n mi)
2	YCP		TRACKAC	east-west coordinate (n mi)
CPX	SITE(1)			see SITE
CPY	SITE(2)			see SITE
CRA20			CNTRLAC,INIT	constant used in equation to determine the distance in a turn when an aircraft is turning from its downwind to its base leg when the intercept angle with the final leg will be 20°
CRA30			CNTRLAC,INIT	constant used in equation to determine the distance in a turn when an aircraft is turning from its downwind to its base leg when the intercept angle with the final will be 30°
CRB20			CNTRLAC,INIT	constant used in equation to determine the distance in a turn when an aircraft is turning from its base to its final leg with a 20° intercept angle
CRB30			CNTRLAC,INIT	constant used in equation to determine the distance in a turn when an aircraft is turning from its base to its final leg with a 30° intercept angle
CRC	TABLE(56)			see TABLE
CROLL	TABLE(55)			see TABLE
CRTINIT	LOGIC(1)			see LOGIC
CRTSCAL	TABLE(13)			see TABLE
CRTTPE			TATHOLD	system name for CRT in use

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
D			COMETA,ETACOMP,TRACKAC	in COMETA and ETACOMP, a temporary for a distance for which an estimated time to fly will be calculated; in TRACKAC, a temporary for a distance
DA			COMETA	temporary for calculated distance (n mi) covered during a change of speed
DAC(I)		(64)	COMMUN,TATINT,TATOPER, TATREST	variable referencing the digital-to-analog converters
DALT			CNTRLAC,TRACKAC	in TRACKAC, temporary for difference between actual altitude and altitude goal; in CNTRLAC, the desired altitude at the end of a path
DECCOR			CNTRLAC	correction for deceleration of airspeed in sequencing area algorithm
DELALT	TABLE(14)			see TABLE
DELAYT			CNTRLAC	delay time (sec) used in a standard or emergency hold; always some multiple of standard or emergency hold time
DELETA			ETACOMP	temporary for the elapsed time in flight calculation for some portion of a path
DELH(I)		(35)	CNTRLAC,TATREST,TRACKAC	amount of flight path error to be applied to the heading of an aircraft in the I position of active traffic
DELSP			TRACKAC	temporary for change of speed rate for a given aircraft under consideration
DELSPD	TABLE(15)			see TABLE
DELT	TABLE(16)			see TABLE
DELX1	TABLE(31)			see TABLE
DELX2	TABLE(32)			see TABLE
DELX3	TABLE(33)			see TABLE

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
DELY1	TABLE(34)			see TABLE
DELY2	TABLE(35)			see TABLE
DELZ			TRACKAC	temporary for climb rate for a particular aircraft under consideration
DGRTD			TATINT	derived constant from the scan increment in sec times gravity constant times a conversion factor from radians to degrees
DIST			TRACKAC	temporary for a distance
DIST1			CNTRLAC	distance in a turn which is calculated in the "compute vector to an ETA point" type controller action
DIST2			CNTRLAC	distance on straight path to an ETA point which is calculated in the "compute vector to an ETA point" type controller action
DK			TRACKAC	amount of heading correction desired on a given scan for a particular aircraft under consideration when the aircraft is flying a DME mode
DKCON	TABLE(19)			see TABLE
DL(I)	ISEQUE(1,15) to ISEQUE(3,15) SEQUE(1,15) to SEQUE(3,15)			see SEQUE
DPRIM			TATREST,TATOPER	temporary
DR1X to DR6Y			TATHOLD	part of the route definition for the background display of the terminal area on the CRT
DSPD	RESTOR(7)			see RESTOR

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
DSPMSG(I)	IDSPMSG(I)	(8)	CRTDSPL	temporary array into which a message is unpacked for J controller
I = 1	IDSPMSG(1) IDSPM1			same as element 1 of ICMSG2(J,1)
2				same as element 2 of ICMSG2(J,1)
3				same as element 3 of ICMSG2(J,1)
4	IDSPMSG(4) IDSPM4			same as element 4 of ICMSG2(J,1)
5				same as element 5 of ICMSG2(J,1)
6	IDSPMSG(6) IDSPM6			same as element 1 of ICMSG3(J,1)
7	IDSPMSG(7) IDSPM7			same as element 2 of ICMSG3(J,1)
8	IDSPMSG(8) IDSPM8			same as element 3 of ICMSG3(J,1)
DST1	RESTOR(11)			see RESTOR
DST2	RESTOR(12)			see RESTOR
DST3	RESTOR(13)			see RESTOR
DTR			INIT, TATINT, TATREST, TATOPER, TRACKAC, CNTRLAC, CRTDSPL	constant to convert from degrees to radians (DTR = .017453)
DX1 DX2 DY1 DY2			CRTDSPL	incremental change of x and y coordinates for backward projected position of aircraft for 1 and 2 scans before current one. Used to portray velocity
D1X to D6Y			TATHOLD	part of the route definition for the background display of the terminal area on the CRT

Table -1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
EHOLDD			CNTRLAC,TATINT	see input data
ENROTA(I)		(2)	CNTRLAC,TATINT	see input data
ERR			TATHOLD	initialization parameter to system routine NAMECRT
ETAPT1			ETACOMP,COMETA	temporary
ETAPT2			ETACOMP,COMETA	temporary
ETASEP	RESTOR(5)			see RESTOR
FETA(I)	ISEQUE(1,16)- ISEQUE(3,16) SEQUE(1,16)- SEQUE(3,16)			see SEQUE
FREQ	TABLE(20)			see TABLE
FSS(I)	LDISI(33)- LDISI(48)	(16)	CNTRLAC,CRTDSPL, TATHOLD,TATOPER,TRACKAC	real-time console function switches for logic flow control in the model
FTNM			TRACKAC,INIT	conversion constant from feet to n mi
FVEL	RESTOR(6)			see RESTOR
G			TATINT,INIT	gravitational constant (G = 32.16 ft/sec^2)
GLIDE(I)		(2)	TATINT,TATOPER	see input data
GNX(I) GNY(I) GSX(I) GSY(I)		(4) (4) (4) (4)	TATHOLD	north and south ILS gate coordinates scaled and translated to CRT units for background display
GTNX(I) GTNY(I) GTSX(I) GTSY(I)		(4) (4) (4) (4)	TATHOLD	north and south ILS gate coordinates in n mi for CRT display use

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
H	TABLE(21)			see TABLE
HDBEG(I)		(35)	TATREST,TRACKAC	initial heading (deg) when aircraft begins a segment with true turn or turn to a heading modes
HDCON	TABLE(59)			see TABLE
HDCOR			TRACKAC	calculated desired heading before applying aircraft bank angle limitations
HDG			CNTRLAC	amount of turn remaining (deg) when aircraft is in 360 hold. Used in calculating initial heading for a standard hold schedule
HDOT			TRACKAC	aircraft turn rate (deg/sec) from aircraft performance tables
HDOWN			TATOPER,TRACKAC, CRTDSPL	heading in deg of cockpit aircraft
HDOWNR			TATOPER	HDOWN in radians
HEADN			TRACKAC	heading (deg) from active traffic array for aircraft being updated this tracking frame
HERR			TRACKAC	error in heading (deg), i.e. difference between desired heading and actual heading
HOLD	LDISI(18)		TATMAIN	console switch to load HOLD overlay
HOLDTEM			CNTRLAC	temporary variable used when resolving conflicts needing holding patterns. Specifically, how much hold time was resolved using standard holds
HOLDTIM			CNTRLAC	total amount of time to lose with holding patterns to resolve a conflict
HT1			CNTRLAC	portion of total time to lose with holding patterns to resolve conflict on next path attributable to beginning end of path ETA conflict

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
HT2			CNTRLAC	portion of total time to lose with holding patterns to resolve conflict on next path attributable to terminal end of path ETA conflict
I			CNTRLAC, COMMUN, CRTDSPL, TATINT, TATPRNT, TATREAD, TATREST, TATHOLD, TATOPER, TRACKAC	temporary
IA(I)	IQUEUE(3)-IQUEUE(4)			see IQUEUE
IAALT			CNTRLAC	actual altitude assigned at terminal end of path, to nearest 1000 ft
IAC			CNTRLAC, TRACKAC	subscript or ID of aircraft in active traffic which is under consideration
IACCEPT(I,J)		(8,2)	CNTRLAC, TATREST	the estimated time on each arrival route when the last aircraft (jet or prop) which has been entered will have flown separation distance in from the perimeter
IACFPER(I)		(3)	TATINT, TRACKAC	see input data and packed array description
IACTN1(I)		(35)	CNTRLAC, COMMUN, TRACKAC	controller action queue: pointer to ITRAF1 array if aircraft is in an arrival or departure queue or to active traffic if aircraft is already in active traffic
IACTN2(I)		(35)	CNTRLAC, COMMUN, TRACKAC	controller action queue: type controller action required
IACTN3(I)		(35)	CNTRLAC, COMMUN, TRACKAC	controller action queue: pointer to which controller action point to use

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
IACTPT			CNTRLAC	stored in packed IETA1 array; it denotes which controller dealt with this aircraft on its last controller action
IADF(I)	INAV(1,2) INAV(10,2)			see INAV packed array
IALT			CRTDSPL	altitude, in nearest hundreds of feet, for CRT display
IALTOP	RESTOR(29) IOPTION(6) OPTION(6)			see RESTOR
IANG			CNTRLAC	subscript of ITANG for determining the angle of intersection with final leg (from base leg) in final sequencing area
IARG			TATINT,TRACKAC	see input data
IARRAY		(variable dimensions)	PAKUNI,UPKUNI	the array to pack into or unpack from--dummy variable used as subroutine call list argument
IARRN			CNTRLAC,TATINT	see input data
IARRS			CNTRLAC,TATINT	see input data
IASKED(I,J)		(35,11)	CNTRLAC,COMMUN,TRACKAC	see packed array description
IASPCON	RESTOR(23)			see RESTOR
IAT			CNTRLAC	temporary
IATAS(I,J)		(35,2)	CNTRLAC,TRACKAC	see packed array description
IATRAF(I,J)	ATRAF(I,J)			see ATRAF
IATRAF1(I)	IATRAF(I,1)			see ATRAF
IATRAF7(I)	IATRAF(I,7)			see ATRAF
IATRAF8(I)	IATRAF(I,8)			see ATRAF

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
IATRAF9(I)	IATRAF(I,9)			see ATRAF
IATRF10(I)	IATRAF(I,10)			see ATRAF
IATRF11(I)	IATRAF(I,11)			see ATRAF
IATRF12(I)	IATRAF(I,12)			see ATRAF
IATRF16(I)	IATRAF(I,16)			see ATRAF
IATRF17(I)	IATRAF(I,17)			see ATRAF
IATRF21(I)	IATRAF(I,21)			see ATRAF
IATRF22(I)	IATRAF(I,22)			see ATRAF
IATRF23(I)	IATRAF(I,23)			see ATRAF
IATRF24(I)	IATRAF(I,24)			see ATRAF
IATRF25(I)	IATRAF(I,25)			see ATRAF
IATSPL	INTEG(71)-INTEG(74)			see INTEG
IBEG	INTEG(70)			see INTEG
IBF(I)		(110)	CRTDSPL,INIT,TATHOLD	buffer for controller messages for display on CRT, also used for aircraft tags on CRT
IBY			CNTRLAC,COMMUN,TRACKAC	parameter (=0) input to system routine WRT218
ICA			CNTRLAC,COMMUN,TATREST, TATOPER,TRACKAC	counter for controller action queue for number of controller actions required this scan
ICACT			CNTRLAC,TATREST,TRACKAC	subscript of next controller action point (from active traffic)
ICACT2	RESTOR(16)			see RESTOR
ICHKPT(I)		(40)	CNTRLAC,TATINT,TRACKAC	see input data and packed array description
ICLASS			CNTRLAC,COMETA,CRTDSPL, ETACOMP,TRACKAC	aircraft performance class; used as subscript to performance tables

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
ICLEAR			COMMUN	number of segment clearances being issued by controller with this message
ICLP1			CNTRLAC	temporary pointer (in sequencing logic) to which active schedule
ICMSG1(I,J)		(8,5)	CNTRLAC,COMMUN	see packed array description
ICMSG2(I,J)		(8,5)	CNTRLAC,COMMUN,CRTDSPL	see packed array description
ICMSG3(I,J)		(8,5)	CNTRLAC,COMMUN,CRTDSPL	see packed array description
ICNGO			CNTRLAC,TATOPER	used as an ASSIGNED GO TO vector when iterations to resolve a conflict may exceed real-time frame time. It allows interruption of logic to update frame counter
ICNT			CNTRLAC	used as counter for number of aircraft needing 360° holds
ICNTF	INTEG(18)			see INTEG
ICNTRA1(I)		(85)	CNTRLAC,TATINT,TRACKAC	see input data and packed array description, controller action point
ICNTRA2(I)		(85)	CNTRLAC,TATINT,TRACKAC	see input data and packed array description, controller action point
ICOMM(I)		(10)	CNTRLAC,COMMUN,CRTDSPL, TATREST	if not zero, counter of how many 4-sec scans needed to deliver message controller is now delivering
ICONTR			CNTRLAC	pointer to which controller is responsible during this controller time frame. Used in final sequencing logic
ICONTRL			CNTRLAC	controller responsible for this action during this controller frame
ICP			CNTRLAC,TRACKAC	current flight leg counter

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
ID	IQUEUE(1)-IQUEUE(2)	(2)		see IQUEUE
IDALT	RESTOR(24) IOPTION(1) OPTION(1)			see RESTOR
IDD(1)			TRACKAC	temporary, used for finding aircraft in active traffic array if one is to be deleted
IDELET	INTEG(66)			see INTEG
IDELT			CNTRLAC,TATINT,TATOPER	integer form of DELT (time in sec between scans)
IDENT			CNTRLAC,COMMUN,CRTDSPL, TRACKAC	aircraft identification number
IDEPN			CNTRLAC,TATINT	see input data
IDEPS			CNTRLAC,TATINT	see input data
IDEST			CNTRLAC	indicator whether departure (=1) or arrival (=2)
IDIAG	INTEG(68)			see INTEG
IDM			COMMUN	temporary for aircraft ID number when searching message queue
IDME(I)	INAV(1,4)-INAV(10,4)			see INAV
IDOTH			CONCHK,SPDCNTR	temporary, ID of other aircraft when checking for "same route conflict"
IDREL	INTEG(25)			see INTEG
IDROUTE(I)		(10)	CNTRLAC,TATINT	see input data and packed array description

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
IDSPD	RESTOR(25) IOPTION(2) OPTION(2)			see RESTOR
IDSPMSG(I)	DSPMSG(I)			see DSPMSG
IDSPM1	DSPMSG(1) IDSPMSG(1)			see DSPMSG
IDSPM4	DSPMSG(4) IDSPMSG(4)			see DSPMSG
IDSPM6	DSPMSG(6) IDSPMSG(6)			see DSPMSG
IDSPM7	DSPMSG(7) IDSPMSG(7)			see DSPMSG
IDSPM8	DSPMSG(8) IDSPMSG(8)			see DSPMSG
IDT			CNTRLAC	temporary; used for deleting an aircraft from departure queue
IDVERG			CNTRLAC	subscript (pointer) to divergent route being tried to resolve conflict
IDVERG2	RESTOR(28) IOPTION(5) OPTION(5)			see RESTOR
IEACH			UPKUNI	in uniform unpack routine, denotes how many elements are packed in each packed word
IEHOP	RESTOR(32) IOPTION(9) OPTION(9)			see RESTOR
IEIGHT			CNTRLAC,COMETA,COMMUN, INIT,TATINT, TRACKAC	symbolic name for the constant 8

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
IEND			PAKSPL	last element to operate on when packing one or more elements into a packed word
IENROT(I,J)		(2,20)	CNTRLAC,COMMUN	arrival queue, i.e., arrival aircraft which is awaiting initial controller action
IENTER			COMMUN	temporary, used for offer time of aircraft from traffic sample
IERR			MAKTAPE	error switch
IETAS(I,J)		(35,2)	CNTRLAC,TRACKAC	see packed array description
IETA1(I,J)		(36,10)	CNTRLAC,SPDCNTR,TRACKAC	see packed array description
IETA2(I,J)		(36,10)	CNTRLAC,SPDCNTR,TRACKAC	see packed array description
IETCOOR(I)		(36)	CNTRLAC,TATINT,TRACKAC	see input data and packed array description
IETT			CNTRLAC	earliest transmit time for a message
IETT2			CNTRLAC	temporary for earliest transmit time for a message developed in the sequencing logic section
IE360(I)		(60)	CNTRLAC,TATREST	counter of number of 120-sec holding pattern increments to allow for delayed holds on a path
IFCMSG1(I,J)		(8,25)	CNTRLAC, COMMUN	see packed array description
IFCMSG2(I,J)		(8,25)	CNTRLAC, COMMUN	see packed array description
IFCMSG3(I,J)		(8,25)	CNTRLAC, COMMUN	see packed array description
IFIVE			CNTRLAC,COMMUN,CRTDSPL, INIT,TATINT,TRACKAC	symbolic name for the constant 5
IFIX(I)	INAV(1,3)-INAV(10,3)			see INAV
IFOUR			CNTRLAC,COMMUN,INIT, TATINT,TATREAD,TRACKAC	symbolic name for the constant 4

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
IGO			CNTRLAC	assigned GO TO vector used in controlling logic flow in the sequencing section
IH			CNTRLAC	temporary
IHEM			CNTRLAC, COMMUN, TATOPER, TRACKAC	hemisphere of aircraft under consideration
IHISIDE			CNTRLAC	temporary, used in final sequencing logic as hemisphere indicator
IHOWN			TATREST, TATOPER, TRACKAC	hemisphere of cockpit
IHPATHE			COMMUN, TATINT	see input data
IHPATHS			CNTRLAC, TATINT	see input data
II			COMMUN, TATPRNT	temporary
IIP1			TATPRNT	temporary
IJKBUFF(I)		(5)	TAATM, TATINT, TATPRNT, TATREAD, TATMAIN	used with system routines, a program control buffer
IK			TATOPER, TRACKAC	temporary
ILEG			CNTRLAC	current flight leg
ILEGM1			CNTRLAC	current flight leg minus one, i.e., previous flight leg
ILIMOP	RESTOR(33) IOPTION(10) OPTION(10)			see RESTOR
ILLIM			CNTRLAC	lower limit of number of aircraft, in an array where a flow-control limit has been applied, before additional separation is removed
ILOC			TRACKAC	temporary
ILTT			COMMUN	latest desirable transmit time for a controller message
IMHERE			CNTRLAC	temporary, used for diagnostic on which logic flow path was taken

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
INAV(I,J)		(10,5)	TATINT,TATOPER,TRACKAC	contains the x,y coordinates, "make good" radii, and standard deviations for the ground-based navigational aid site. See packed array description and input data for details
J = 1	IVOR(I)	(10)	TRACKAC	VORs
2	IADF(I)	(10)	TRACKAC	ADFs
3	IFIX(I)	(10)	CRTDSPL,TRACKAC	FIXes
4	IDME(I)	(10)	TRACKAC	DMEs
5	IXILS(I)	(10)	TATOPER,TRACKAC	ILSs
INC	INTEG(8)			see INTEG
INCC	INTEG(9)			see INTEG
INCP			TATPRNT	temporary
IND			CNTRLAC,TATHOLD	temporary--used as switch in sequencing logic
INDEX			TATINT	temporary
INDR			CNTRLAC	indicator of whether or not aircraft will fly over the inner fix in the sequencing area
INDSPD	RESTOR(36)			see RESTOR
INDT			CONCHK	logic control switch
INEXT			CNTRLAC	pointer to traffic sample for next aircraft waiting for departure or arrival clearance
INFO4			TRACKAC	temporary
ININE			CNTRLAC,INIT,TATINT,TATREAD,TRACKAC	symbolic name for the constant 9
INITRD	LOGIC(8)			see LOGIC

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
INTEG(I)		(80)		array of integer variables that can be called up on the digital display on the console fairly easily
I = 1	ISCAL		INIT,TATHOLD	scaling factor for CRT display
2	IPL		TAATM,TATINT,TATMAIN	denotes primary overlay
3	ISL		TAATM,TATINT,TATMAIN	denotes secondary overlay
4	N		COMMUN,MAKTAPE,TATOPER PAKUNI,TRACKAC,RANDU	temporary in COMMUN,MAKTAPE,PAKUNI,RANDU. in TRACKAC, used as pointer to active traffic aircraft being tracked--initialized in operate
5				not used at present
6	NFMSGD		CNTRLAC,TATOPER	counter of messages developed this frame to update frame counter if necessary to maintain real-time synchronization
7	KRET		INIT,CNTRLAC	used with NFMSGD as the threshold count of messages developed this frame whereby any more would possibly cause time synchronization loss
8	INC		CNTRLAC,TATOPER,TATREST, COMMUN	frame counter for real-time frames for each scan
9	INCC		TATOPER,CRTDSPL,INIT	number of aircraft for which display calculations are going to be done this frame
10	INUM		COMMUN,TATREST	pointer to traffic sample array for next aircraft to be considered for entry into arrival or departure queues
11	IO		INIT,TATOPER	temporary

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
INTEG(I)				
I = 12	IOUT		TATINT,TATPRNT,TATOPER, TRACKAC,COMMUN,CNTRLAC, INIT	used as indicator of type of output data to write on real-time file
13	IOPT		TRACKAC,INIT,CRTDSPL, TATHOLD	option switch on which CRT plot option to use (e.g., relative or standard)
14				not used at present
15				not used at present
16				not used at present
17	IR		TATREST,RANDU,TRACKAC	kernel to random number generator, initialized to small odd integer, then updated after each RANDU call
18	ICNTF		CNTRLAC,TATOPER	real-time frame counter, includes extra time frame used to keep from losing time synchronization
19	IRUN		TATREST,INIT,TATINT, TATPRNT,TATHOLD	real-time run number
20				not used at present
21	ISW2		TATREST,TRACKAC	switch that causes looping until the operator has entered on the console the aircraft ID number of an aircraft that needs to be deleted
22	ISW3		TATREST,TRACKAC	switch that causes looping until the operator has entered on the console the aircraft ID number of the aircraft that the cockpit is supposed to follow
23				not used at present
24				not used at present

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
INTEG(I)				
I = 25	IDREL		TATREST,TRACKAC	ID of aircraft that is selected as the aircraft to put the relative coordinate system on for relative display
26				not used at present
27	KKK		TATOPER,CRTDSPL,CNTRLAC	temporary
28	K1		CRTDSPL	temporary
29	K2		CRTDSPL	temporary
30	MAXAC		INIT,TATREST,CNTRLAC	maximum number of aircraft accepted into active traffic; value is set to agree with storage dimensions reserved
31				not used at present
32				not used at present
33	NCOM		CNTRLAC,COMMUN,INIT, TATOPER	denotes which operate frames are to be used for channel communications
34	NCONTRL		INIT,TATOPER	denotes when all controller action frames have been completed in current scan
35	MAXQUE		COMMUN,INIT	upper limit set on how many aircraft can be waiting in arrival or departure queues
36	NFRAM		INIT,TATINT,TATOPER	number of time frame before update of scan counter. Usage is to override this to run in fast time when desirable
37	NTRAF		TATINT,TATREAD	see input data
38	NTRAK		COMMUN,INIT,TATOPER	denotes which operate frames are to be used for aircraft position updating (tracking)
39				not used at present
40	IT		CNTRLAC,COMMUN,CRTDSPL, TATINT,TATPRNT,TATREST, TATOPER	scan counter

3

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
INTEG(I)				
I = 41	NSCAN		CNTRLAC,COMMUN,CONCHK, CRTDSPL,TATINT,TATPRNT, TATREST,TATOPER,SPDCNTR, TRACKAC	number of aircraft in active traffic
42				not used at present
43				not used at present
44				not used at present
45	ISTK		CRTDSPL,INIT,TATOPER, TRACKAC	switch to indicate if stick (cockpit) condition is on (ISTK≠0)
46 to 54	MSG(I)	(9)	CNTRLAC,TATINT	see input data
55 to 65				not used at present
66	IDELET		TRACKAC	aircraft number (input from real-time console) to be deleted from active traffic
67	ISTOP		TATOPER	variable set from real-time console--used to permit stopping on any frame
68	IDIAG		TATPRNT	when switch is set, i.e., IDIAG≠0, schedule will be printed with active traffic
69				not used at present
70	IBEG		COMMUN,TATREST,TATHOLD, TATOPER,TRACKAC,PAKSPL	temporary
71 to 74	IATSPL(I,J)	(4,1)	TATREST,TATHOLD,TATOPER	special attribute table for unpacking from core storage
INUM	INTEG(10)			see INTEG
INUM1			COMMUN	position in traffic sample of first aircraft in takeoff queue 1 (ITOQUE(1,1))

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
INUM2			COMMUN	position in traffic sample of first aircraft in takeoff queue 2 (ITOQUE(2,1))
IO	INTEG(11)			see INTEG
IOBETA			CONCHK,SPDCNTR	subscript of ETA at beginning end of path for "other aircraft"
IOEETA			CONCHK,SPDCNTR	subscript of ETA at terminal end of path for "other aircraft"
IONE			CNTRLAC,COMMUN,CONCHK, CRTDSPL,INIT,TATINT, TATREAD,TATOPER,SPDCNTR, TRACKAC	symbolic name for constant 1
IOPATH			CONCHK,SPDCNTR	subscript of NNPATH for "other aircraft"
IOPT	INTEG(13)			see INTEG
IOPTION(I)	OPTION(I) RESTOR(24) RESTOR(33)			see RESTOR
IOPTN(I)		(55)	CNTRLAC,TATINT	see input data and packed array description
IOTHID			SPDCNTR,CONCHK	temporary used in searching ETA array for ID of "other aircraft"
IOUT	INTEG(12)			see INTEG
IPA			CNTRLAC	for runway conflict check for departure clearance, it is the subscript of ETA array where arrivals are registered for other runway
IPACK			TATINT	temporary for loop index
IPAPATH			CNTRLAC,COMMUN,TATINT	see input data--attribute table for packed words IATRF21 and IATRF23

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
IPASKED(I,J)		(4,7)	CNTRLAC,COMMUN,TATINT, TRACKAC	input data--attribute table for packed array IASKED
IPASSAC			CONCHK,TATREST,SPDCNTR	counter for number of aircraft that can be passed in conflict check ETA array
IPCHKPT(I,J)		(4,6)	CNTRLAC,TATINT,TRACKAC	input data--attribute table for packed array ICHKPT
IPCNTR1(I,J)		(4,6)	CNTRLAC,TATINT,TRACKAC	input data--attribute table for packed array ICNTRA1
IPCNTR2(I,J)		(4,2)	CNTRLAC,TATINT,TRACKAC	input data--attribute table for packed array ICNTRA2
IPD			CNTRLAC	for runway conflict check for departure clearance, it is the subscript of ETA array where departures are registered for other runway
IPDRT(I,J)		(4,5)	CNTRLAC,TATINT	input data--attribute table for packed array IDROUTE
IPERFR(I,J)		(3,7)	TATINT	see input data and packed array description
IPERFR2(I)	IPERFR(I,2)	(3)	CNTRLAC,TRACKAC	see input data and packed array description,IPERFR
IPERFR3(I)	IPERFR(I,3)	(3)	CNTRLAC,TRACKAC	see input data and packed array description,IPERFR
IPERFR4(I)	IPERFR(I,4)	(3)	CNTRLAC,TRACKAC	see input data and packed array description,IPERFR
IPERFR5(I)	IPERFR(I,5)	(3)	CNTRLAC,TRACKAC	see input data and packed array description,IPERFR
IPERFR6(I)	IPERFR(I,6)	(3)	COMETA,TRACKAC	see input data and packed array description,IPERFR
IPERFR7(I)	IPERFR(I,7)	(3)	CNTRLAC,TRACKAC	see input data and packed array description,IPERFR
IPERFS(I,J)		(4,8)	CNTRLAC,TATINT,TRACKAC	see input data and packed array description
IPERFS1(I)	IPERFS(I,1)	(4)	CNTRLAC,TRACKAC	see input data and packed array description,IPERFS
IPERFS2(I)	IPERFS(I,2)	(4)	CNTRLAC,ETACOMP,TRACKAC	see input data and packed array description,IPERFS

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
IPERFS3(I)	IPERFS(I,3)	(4)	CNTRLAC,ETACOMP,TRACKAC	see input data and packed array description,IPERFS
IPERFS4(I)	IPERFS(I,4)	(4)	TATINT	see input data and packed array description,IPERFS
IPERFT(I)		(3)	TATINT,TRACKAC	see input data and packed array description
IPETA1(I,J)		(4,4)	CNTRLAC,CONCHK,TATINT SPDCNTR,TRACKAC	input data--attribute table for packed array IETA1
IPETA2(I,J)		(4,6)	CNTRLAC,CONCHK,TATINT, SPDCNTR	input data--attribute table for packed array IETA2
IPETC(I,J)		(4,4)	CNTRLAC,TATINT,TRACKAC	input data--attribute table for packed array IETCOOR
IPFTR(I)		(3)	CNTRLAC,TATINT,TRACKAC	see input data and packed array description
IPL	INTEG(2)			see INTEG
IPLACE			UPKUNI	temporary
IPLIMAC(I,J)		(4,7)	TATINT,CNTRLAC	input data--attribute table for packed array LIMAC
IPMSG2(I,J)		(4,6)	CNTRLAC,COMMUN,CRTDSPL, TATINT	input data--attribute table for packed arrays ICMSG2 and IFCMSG2
IPMSG3(I,J)		(4,3)	CNTRLAC,COMMUN,CRTDSPL, TATINT	input data-attribute table for packed arrays ICMSG3 and IFCMSG3
IPNPATH(I,J)		(4,7)	CNTRLAC,COMMUN,TATINT TRACKAC	input data--attribute table for packed array NNPATH
IPNSITE(I,J)		(4,4)	TATINT,TATOPER,TRACKAC	input data--attribute table for packed array INAV
IPOPPA			TATINT	see input data
IPOPTN(I,J)		(4,10)	CNTRLAC,TATINT	input data--attribute table for packed array IOPTN
IPOPUP(I)	POPUP(I) TABLE(60)-TABLE(72)			see TABLE
IPSKED		(4,8)	CNTRLAC,COMMUN,TATINT, TRACKAC	input data--attribute table for packed array ISKED

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
IPTAS(I,J)		(4,5)	TATINT,TRACKAC	input data--attribute table for packed arrays IETAS and IATAS
IPTRAF1(I,J)		(4,4)	CNTRLAC,TATINT,TATREAD, TRACKAC	input data--attribute table for packed array ITRAF1
IPTRAF2(I,J)		(4,9)	CNTRLAC,COMMUN,TATINT, TATREAD,TATREST	input data--attribute table for packed array ITRAF2
IQUEUE(I)		(4)	CNTRLAC	variable name for total set of departure and arrival queues; this name optionally used by flow control option when checking count of aircraft in departure or arrival queues
I = 1	ID(I)	(2)	CNTRLAC,COMMUN,TATREST, TATOPER,TRACKAC	departure queue counter, contains count of aircraft in departure queue who are waiting for controller action
3	IA(I)	(2)	CNTRLAC,COMMUN,TATREST, TATOPER	arrival queue counter, contains count of aircraft in arrival queue whose offer time is now or has passed
IR	INTEG(17)			see INTEG
IRCLR			CNTRLAC	segment clearances being removed when a message is being deleted for an aircraft
IROUTE			CNTRLAC	arrival route number used as subscript to IACCEPT array
IRSEP			CNTRLAC	subscript to separation array SEPP for final sequencing logic
IRUN	INTEG(19)			see INTEG
IS			CNTRLAC	temporary, used as subscript to ETA array to check for other aircraft needing 360s
ISA			CNTRLAC	for runway conflict check for departure clearances, it is the subscript of ETA array where arrivals are registered for same runway
ISAC(I)	ISEQUE(1,16)- ISEQUE(3,16), SEQUE(1,16)-			see SEQUE

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
ISAV			COMMUN,TRACKAC	temporary used as subscript in COMMUN and as a loop limit in TRACKAC
ISCAL	INTEG(1)			see INTEG
ISCLAS(I)	ISEQUE(1,3)-ISEQUE(3,3), SEQUE(1,3)-SEQUE(3,3)			see SEQUE
ISD			CNTRLAC	for runway conflict check for departure clearances, it is the subscript of ETA array where departures are registered for same runway
ISDST(I)		(17)	CNTRLAC,TATINT	see input data and packed array description
ISEG1			CNTRLAC	temporary, used as subscript in popup schedule logic
ISEG2			CNTRLAC	temporary, used as subscript in popup schedule logic
ISEQ			CNTRLAC,TRACKAC,TATREST	count of aircraft in sequencing queue
ISEQM1			CNTRLAC	numerically, one less than number of aircraft in sequencing queue
ISEQUE(I,J)	SEQUE(I,J)			see SEQUE
ISEVEN			CNTRLAC,COMMUN,INIT, TATINT,TRACKAC	symbolic name for the constant 7
ISHFT			UPKUNI	temporary
ISHIFT			PAKSPL and UPKSPL	shift to align word to unpack first element wanted--from row 4 of attribute tables
ISHOP	RESTOR(31) IOPTION(8) OPTION(8)			see RESTOR

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
ISIDE(I)	ISEQUE(1,2)-ISEQUE(3,2), SEQUE(1,2)-SEQUE(3,2)			see SEQUE
ISIGN			PAKSPL and UPKSPL	indicator whether element wanted in packed word is signed (=1)--from row 2 of attribute tables
ISIX			CNTRLAC,COMMUN,CONCHK, ETACOMP,INIT,TATINT, TATREST,SPDCNTR,TRACKAC	symbolic name for the constant 6
ISKED(I)		(85)	CNTRLAC,COMMUN,TATINT, TRACKAC	see input data and packed array description
ISL	INTEG(3)			see INTEG
ISPD			CRTDSPL	aircraft speed in terms of knots for tags on CRT display
ISPDOP	RESTOR(30) IOPTION(7) OPTION(7)			see RESTOR
ISPMSG			CNTRLAC	indicator set if a speed change message is to be generated
ISTARTA(I)		(21)	COMMUN,TATINT	see input data
ISTK	INTEG(45)			see INTEG
ISTOP	INTEG(67)			see INTEG
ISUB			COMMUN,TRACKAC	temporary, used as ETA array subscript--in COMMUN is specifically subscript of conflict checkpoint ETA for this message
ISUBCK	RESTOR(14)			see RESTOR
ISUBN	RESTOR(18)			see RESTOR

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
ISUBO	RESTOR(19)			see RESTOR
ISUBOP	RESTOR(15)			see RESTOR
ISWHOLD			CNTRLAC,TRACKAC	location of element in IETCOOR packed word for a switch which is turned on if an aircraft is scheduled for standard holding pattern over the conflict checkpoint
ISWITCH			CNTRLAC	subscript to SWITCH array which contains on-off switches for flow-control limits logic
ISW2	INTEG(21)			see INTEG
ISW3	INTEG(22)			see INTEG
ISW4			TATOPER,TRACKAC	switch used with cockpit aircraft
IT	INTEG(40)			see INTEG
ITANG(I)		(4)	CNTRLAC,TATINT	see input data
ITAPE			MAKTAPE	dummy variable in subroutine argument list --tape unit number
ITEMP			CNTRLAC,COMMUN,TATINT, TATREAD,TRACKAC	temporary scalar integer
ITEMPA(I)	TEMPA(I)	(10)	CNTRLAC,TRACKAC	temporary integer mode array
ITEMPA1			COMMUN	temporary
ITEMPA2			COMMUN	temporary
ITEMPA3			COMMUN	temporary
ITEMPS			CNTRLAC,TRACKAC	temporary scalar integer
ITEMP2			CNTRLAC,TRACKAC	temporary scalar integer
ITEN			CNTRLAC,INIT,TATINT, TRACKAC	symbolic name for the constant 10

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
ITEST			SPDCNTR	temporary for logic control
ITHREE			CNTRLAC,COMMUN,CONCHK, CRTDSPL,INIT,TATREST, SPDCNTR	symbolic name for the constant 3
ITIME			TRACKAC	temporary, for type time in CP mode for time
ITL			TATPRNT,TATREST	switch used in print routine to determine if data for a new scan has been encountered
ITOQUE(I,J)		(2,20)	CNTRLAC,COMMUN,TRACKAC	takeoff queue containing ID number of aircraft which are awaiting controller action for departure clearances
ITRAF1(I)		(102)	CNTRLAC,TATINT,TATREAD, TRACKAC	see input data and packed array description
ITRAF2(I)		(102)	CNTRLAC,COMMUN,TATINT, TATREST,TATREAD,TRACKAC	see input data and packed array description
ITRIB(I,J)		(4,1)	PAKSPL and UPKSPL	attribute table, variable dimensions, dummy variable in subroutine call argument list
ITRY			CNTRLAC,CONCHK,ETACOMP, TATREST,SPDCNTR	temporary indicator and switch for logic flow control
ITRYDVG			CNTRLAC	switch to indicate whether divergent route option was tried to resolve conflict
ITRYEH			CNTRLAC	switch to indicate whether 360 (emergency hold) option was tried to resolve conflict
ITRYSH			CNTRLAC	switch to indicate whether standard hold pattern option was tried to resolve conflict
ITRYSPD			CNTRLAC	switch to indicate whether speed control option was tried to resolve conflict
ITRYVEC			CNTRLAC	switch to indicate whether alternate vector option was tried to resolve conflict

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
ITT			CNTRLAC	temporary for time in turn
ITWO			CNTRLAC,COMMUN,CONCHK, INIT,TATINT,TATOPER, SPDCNTR,TRACKAC	symbolic name for the constant 2
ITYPCL			CNTRLAC,TRACKAC	type of controller action
ITYPE			CNTRLAC	aircraft type (prop or jet)
IULIM			CNTRLAC	upper limit of number of aircraft desired in an array being limited by flow control; if exceeded, separation requirements are increased
IVARBUFF(I)		(5)	TATINT	buffer to system routine
IVARSPD			CNTRLAC	switch to indicate if speed variation has been used
IVECOP	RESTOR(27) IOPTION(4) OPTION(4)			see RESTOR
IVECT	RESTOR(26) IOPTION(3) OPTION(3)			see RESTOR
IVECT2			CNTRLAC	subscript to NNPATH for next possible alternate vector--no more if zero
IVOR(I)	INAV(1,1)- INAV(10,1)			see INAV
IWARR			TRACKAC	subscript of arrival ETA array used for touchdown CP mode
IWARRAY			CNTRLAC	subscript to which array (ETA or queue) the flow-control limit applies

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
IWCHAO			CNTRLAC	used with IDALT as subscripts to the altitude option array ALTOPT for altitude being considered
IWETA			CNTRLAC	temporary for ETA array subscript
IWH			TRACKAC	temporary used as subscript for unpacking IETAS and IATAS arrays
IWHCH			CNTRLAC,UPKUNI	temporary used to point to which packed word or element on which to operate
IWHCHM1			CONCHK,SPDCNTR	pointer used while checking aircraft behind and <u>ahead</u> during conflict check
IWHCHP1			CONCHK	pointer used while checking aircraft <u>behind</u> and ahead during conflict check
IWHCNTR			CNTRLAC	subscript for controller who generated path and message, if any
IWHICH			CNTRLAC,CONCHK,TATREST, SPDCNTR	temporary subscript of aircraft position in conflict checkpoint array (ETA)
IWHP1			CNTRLAC,SPDCNTR	temporary subscript of conflict checkpoint array (ETA), specifically, IWHICH+1
IWHSPD			CNTRLAC	counter (subscript) of which alternate speed variation now being tried to resolve conflict
IWID			CNTRLAC	temporary for aircraft ID in message arrays
IWORD			TATINT,PAKSPL,UPKSPL, UPKUNI	temporary, used for packing and unpacking elements
IWORK			PAKUNI	temporary, used for packing elements into a packed word
IWPATH			COMMUN	switch to indicate whether message applies to present path (1) or future path (2)
IXCESS			PAKUNI	count of excess bits when elements do not fill a packed word--used to position elements uniformly

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
IXI			CNTRLAC,COMMUN,TRACKAC	argument list variable to system routine
IXILS(I)	INAV(1,5)-INAV(10,5)			see INAV
IXRET			UPKUNI	temporary scalar integer
IY			RANDU,TRACKAC	"update of kernel" variable for the calls to random number generator routine
J			CNTRLAC,COMMUN,CRTDSPL TATINT,TATREST,TATOPER, TRACKAC	input data in TATINT, but used as temporary elsewhere
JGO			CNTRLAC	assigned go to switch for accepting aircraft into active traffic
JJ			CNTRLAC,COMMUN,CRTDSPL, TATINT,TRACKAC	input data in TATINT, but used as temporary elsewhere
JJJ			CNTRLAC,COMMUN,TRACKAC	temporary
JJM1			CNTRLAC	temporary
JJP1			CNTRLAC	temporary

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
JL			CNTRLAC,CONCHK,SPDCNTR	temporary
JLP1			CONCHK,SPDCNTR	temporary
JOE			CNTRLAC,CONCHK	temporary--used as subscript
JP1			CNTRLAC	temporary
JS			CNTRLAC	temporary
JTEMP			CNTRLAC,CONCHK,SPDCNTR, TRACKAC	temporary
JTEMPA(I)		(10)	CNTRLAC	temporary integer mode array
JWORK			PAKUNI	temporary
JX			COMMUN	temporary
JXX			COMMUN	temporary
K			CNTRLAC,COMMUN,TATINT, TATREST,PAKSPL/UPKSPL, PAKUNI,TRACKAC	temporary
KADF			TRACKAC	temporary, used as subscript to ADF site
KALTOP			CNTRLAC	temporary (for IWCHAO)--subscript of optional altitude being considered if option is available
KDATA			PAKUNI	temporary
KDME			TRACKAC	temporary--subscript of DME site

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
KGO			CNTRLAC	assigned GO TO variable used when developing messages
KILS			TATOPER,TRACKAC	temporary--used as subscript of ILS site
KK			CNTRLAC,COMMUN,PAKUNI, TRACKAC	temporary
KKK	INTEG(27)			see INTEG
KPACK			TATINT	temporary--used as count of individual type navaid sites while packing
KP3			CNTRLAC	temporary--used when transferring schedule from NNPATH to active schedule
KRET	INTEG(7)			see INTEG
KTEMP			CNTRLAC,CONCHK,SPDCNTR	temporary
KTYPE			TATINT	temporary--used in argument list to system routine
KVOR			TRACKAC	temporary--used as subscript to VOR site
KWORD			PAKSPL,UPKUNI	temporary
K1	INTEG(28)			see INTEG
K2	INTEG(29)			see INTEG
K3			CNTRLAC	temporary

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
L			COMMUN,TATOPER,TRACKAC, PAKUNI	used as temporary except in TRACKAC, where it is a counter for aircraft being tracked this scan
LDISI(I)		(108)	TAATM,TATINT,TATMAIN, PAKSPL	array of LOGICAL discrete inputs (switches) allowing interaction with the real-time simulation from the console
LDISO(I)		(196)	CRTDSPL,TATINT,TATPRNT,	array of LOGICAL discrete outputs (lights or logical switches) to set and test switches in the real-time simulation
LEMG(I)		(5)	CNTRLAC,TATREST	array of pointers to positions in an ETA array for all aircraft behind aircraft under consideration requiring emergency 360s to solve a conflict
LENGTH			TATINT,PAKSPL,PAKUNI, UPKUNI	temporary for how many bits each element requires in the packed word--appears in argument list for uniform pack routines PAKUNI/UPKUNI and as row 1 of attribute table for PAKSPL/UPKSPL
LIMAC(I)		(10)	CNTRLAC,TATINT	see input data and packed array description
LIMGO			CNTRLAC	assigned GO TO variable to control logic in flow control section of routine
LIND			CNTRLAC	indication of whether an attempt to resolve conflict with holding pattern has already been tried
LL			CNTRLAC,TRACKAC	temporary
LLL			CNTRLAC,COMMUN,TRACKAC	temporary--used as subscript when assigning active schedule or determining which is most recently assigned schedule
LLLM1			CNTRLAC	temporary--used with LLL as pointer to active schedule

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
LLLP1			CNTRLAC	temporary--used with LLL as pointer to active schedule
LLLP2			CNTRLAC	temporary--used with LLL as pointer to active schedule
LLLP3			CNTRLAC	temporary--used with LLL as pointer to active schedule
LOGIC(I)		(20)	TATINT,INIT	an array of logical typed variables used for console interaction with the real-time simulation system
I = 1	CRTINIT		TATHOLD,INIT,TATOPER	logical variable which when turned off prohibits multiple passes through the CRT initialization logic
2 to 6				not currently being used
7	WRTTPE		INIT,TATPRNT,TATMAIN	logical variable used as a switch for additionally writing BCD tape of outputs normally printed
8	INITRD		INIT,TATREST,TATOPER	logical variable used as switch to bypass the initialization logic except for the first time through when in reset mode
9	MSGTYP		COMMUN,INIT	logical variable used as switch to output controller messages on the on-line typewriter
10				not currently being used
11	PLOTSW		INIT,TATHOLD,TATOPER	logical variable used as a switch to prevent multiple passes through program logic when writing tape images for CALCOMP plotter
12 to 20				not currently being used

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
LPACK			TATINT	temporary
LTEMP			CNTRLAC,TRACKAC	temporary
LWORD			TATINT,TATHOLD,TATOPER, PAKSPL	temporary
LWORD1			TATINT,TATREAD	temporary
LWORD2			TATINT,TATREAD	temporary
M			CNTRLAC	temporary
MASK			PAKSPL/UPKSPL,UPKUNI	field of bits generated and used for packing and unpacking elements into or from packed words
MAXAC	INTEG(30)			see INTEG
MAXQUE	INTEG(35)			see INTEG
MF			CNTRLAC,TAATM,TATPRNT, TRACKAC	I/O unit assigned for day-file output
MM			CNTRLAC	temporary
MMM			CNTRLAC	temporary
MODE			TRACKAC	temporary for element number 1 in ISKED or IASKED--see ISKED packed data description
MODECP			TRACKAC	temporary for element 5 in IASKED and element 6 in ISKED. See ISKED packed data description

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
MODINC			TATOPER	modulo 4 counter for cockpit logic
MOVE			CNTRLAC	temporary used as subscript when placing aircraft back into controller action array IACTN1, IACTN2 and IACTN3 for re-look
MSG(I)	INTEG(46)-INTEG(54)			see INTEG
MSGT			COMMUN	temporary for message type
MSGTYP	LOGIC(9)			see LOGIC
MTEMP			CNTRLAC	temporary
MTYPE			CRTDSPL	temporary used as computed GO TO for message type to be used in developing the CRT display
N	INTEG(5)			see INTEG
NADC			TATINT	number of analog-to-digital converters for set-up request
NADF	NNAV(2)			see NNAV
NALTOP			TATINT	see input data
NALTSKD			TATINT	see input data
NARR			TATINT	see input data
NAR1 through NAR8			TATHOLD	switches used for plotting arrival Victor airways displays on CRT

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
NASPACE(I)		(4)	CNTRLAC,TATREST	number of aircraft in the airspace conflict arrays ASPACE1, ASPACE2, ASPACE3
NCACT			TATINT	see input data
NCHK			TATINT	see input data
NCLRD			CNTRLAC	number of aircraft which have priority because they have been cleared through this point and the next one
NCMSG(I)		(10)	CNTRLAC,COMMUN,TATREST	number of current messages in Ith controller's message queue
NCMSGT			CNTRLAC,COMMUN	temporary, number of current messages in the queue for this controller
NCOM	INTEG(33)			see INTEG
NCONTRL	INTEG(34)			see INTEG
NCP			TATINT,TATPRNT,TRACKAC	see input data
NDAC			TATINT	number of digital-to-analog converters for set-up request
NDEPR			CNTRLAC,TATINT,TRACKAC	see input data
NDME	NNAV(4)			see NNAV
NDRTE			TATINT	see input data
NDR1 through NDR6			TATHOLD	switches used for plotting departure Victor airways displays on CRT
NEMG			CNTRLAC,TATREST	number of aircraft requiring emergency (immediate) 360 holding patterns
NETA(I)		(36)	CNTRLAC,CONCHK,TATREST, TATOPER,SPDCNTR,TRACKAC	array of counts of aircraft in each ETA array

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
NFIX	NNAV(3)			see NNAV
NFMSG(I)		(10)	CNTRLAC,COMMUN,TATREST	number of future messages in Ith controller's message queue
NFMSGD	INTEG(6)			see INTEG
NFMSGT			CNTRLAC,COMMUN	temporary--number of future messages in the queue for this controller
NFRAM	INTEG(36)			see INTEG
NFRAMD			TATOPER	number of calls to display CRT each frame
NFRAMF			TATHOLD,TATOPER	frequency with which CRT is updated for all aircraft
NGO			TRACKAC	for deleting an aircraft from active traffic; assigned GO TO switch used in deleting an aircraft from one or two ETA arrays
NHOLD			TATMAIN	assigned GO TO mode control "HOLD" return to top of HOLD logic
NHOLDP(I)		(35)	CNTRLAC,COMMUN,CONCHK, CRTDSPL,SPDCNTR,TRACKAC	number of segments remaining until the Ith aircraft in active traffic completes a standard holding pattern if non-zero
NILS	NNAV(5)			see NNAV
NINTEG			TATINT	number of integer tables requested in set-up request
NLDISI			TATINT	number of LDISIs (logical discrete inputs) for set-up request
NLDISO			TATINT	number of LDISOs (logical discrete outputs) for set-up request
NLIM			TATINT	number of flow-control optional limits arrays to read in
NLOGIC			TATINT	size of LOGIC variable table in set-up request

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
NNA(I)		(35)	CNTRLAC,COMMUN,TRACKAC	number of segments currently in active schedule of Ith aircraft
NNAT			TRACKAC	temporary
NNAV(I)		(5)	TATINT	number of navigational aids per type of aid
I = 1	NVOR		TATINT	see input data
2	NADF		TATINT	see input data
3	NFIX		TRACKAC,TATINT	see input data
4	NDME		TATINT	see input data
5	NILS		TATINT	see input data
NNPATH(I)		(50)	CNTRLAC,COMMUN,TATINT, TRACKAC	see input data and packed array description
NOM			CNTRLAC	temporary switch used to save nominal path when trying another option
NOPER			TATMAIN	assigned GO TO variable to return control of program to top of operate section logic
NOPT			TATINT	see input data
NOUT(I)		(20)	CNTRLAC,TATINT,TATPRNT, TATOPER,TRACKAC	temporary array used to output schedules and sequencing queues
NPRINT			TATMAIN	assigned GO TO variable to return control to top of print section of logic
NPTH			TATINT	see input data
NREAD			TATMAIN	assigned GO TO variable to return control to top of read section of logic
NRESET			TATMAIN	assigned GO TO variable to return control to top of reset section of logic
NS			DASHCIR	temporary for controlling dashed lines in drawing a circle

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
NSCAN	INTEG(41)			see INTEG
NSEGM	RESTOR(10)			see RESTOR
NSKED			TATINT	see input data
NSPDSTD			TATINT	see input data
NSPDW			CNTRLAC,TATINT	temporary for number of packed words for each row in speed performance arrays
NSTART			TATINT	see input data
NTABLE			TATINT	number of locations in TABLE array needed on set-up request
NTEMP			CNTRLAC,TRACKAC	temporary
NTERM			TATMAIN	assigned GO TO variable to return control to top of terminate section of logic
NTRAF	INTEG(37)			see INTEG
NTRAK	INTEG(38)			see INTEG
NTWO	INTEG(23)			see INTEG
NTYPE			TATINT	see input data
NUMPATH	RESTOR(2)			see RESTOR
NUMSC	INTEG(39)			see INTEG
NVOR	NNAV(1)			see NNAV
NWORDS			PAKUNI	temporary--number of consecutive computer words needed to pack this array
N360			CNTRLAC	integer number of 360s being considered for an aircraft to resolve a conflict

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
OALT	RESTOR(3)			see RESTOR
OLDR(I)		(35)	CNTRLAC,TRACKAC	old range of aircraft from navaid sites on previous scan
OLFN			TAATM,TATINT	actial file name where overlays reside
ONOFF			CNTRLAC	temporary for SWITCH(I)
OPER	LDISI(17)		TATMAIN	console switch to take real-time simulation to operate mode
OPTION(I)	IOPTION(I) RESTOR(24)-RESTOR(33)			see RESTOR
OSCAR			TATINT	literal argument to NM218 system routine
OSPD	RESTOR(34)			see RESTOR
OTHAA			CONCHK,SPDCNTR	altitude of "other aircraft" at terminal end of path being considered
OTHETA			CONCHK,SPDCNTR	ETA of "other aircraft" at conflict point being considered
OTHOA			CONCHK,SPDCNTR	altitude of "other aircraft" at beginning of path being considered
OTHSETA			CONCHK,SPDCNTR	ETA of "other aircraft" at separation distance from conflict point being considered
OTHVEL			CONCHK	velocity for "other aircraft" at conflict checkpoint being considered
OUT(I)		(26)	CNTRLAC,COMMUN,TATINT, TATPRNT,TRACKAC	temporary array used to output message, active traffic ETA information, etc.

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
PABS			TATOPER	three-dimensional range from runway in n mi
PASEP	TABLE(23)			see TABLE
PDSEP	TABLE(24)			see TABLE
PERSPLT	LOGIC(10)			see LOGIC
PHI			TRACKAC	bank angle needed to correct aircraft heading (maximum value permitted is 30°)
PHIK	TABLE(25)			see TABLE
PI			INIT	the constant π, where 2π radians - 360°
PID2			INIT	one half PI, i.e., $\pi/2$
PITCH			TATHOLD,TATOPER	cockpit aircraft pitch angle, in degrees, used for runway display
PITCHC			TATOPER	temporary used as cosine of PITCH angle
PITCHS			TATOPER	temporary used as sine of PITCH angle
PI20			INIT	20 times PI, i.e., 20π
PLOTSW	LOGIC(11)			see LOGIC
POPUP	IPOPUP(I) TABLE(60)-TABLE(72)			see TABLE
PPMX(I)		(3)	CRTDSPL,TATINT	see input data
PPMY(I)		(3)	CRTDSPL,TATINT	see input data
PPX1			CRTDSPL	used in drawing line segments on the CRT
PPX2			CRTDSPL	same as PPX1
PPY1			CRTDSPL	same as PPX1
PPY2			CRTDSPL	same as PPX1
PSI			XFORM	used as dummy variable in subroutine argument list for aircraft heading

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
PSIAC			TRACKAC	temporary: direction, in degrees, of a navigational aid site from present position of aircraft
PSIR			TRACKAC	temporary: defines which radial (or pseudo-radial) of a navigational aid site to use
PSIREL	TABLE(10)			see TABLE
PTEMP			TRACKAC	temporary: used in calculating PSIAC
PX			TATOPER	temporary: used in cockpit display for runways' position relative to aircraft
PY			TATOPER	temporary: same as PX
PYP			TATOPER	temporary: same as PX
PZ			TATOPER	temporary: same as PX
PZP			TATOPER	temporary: same as PX

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
R			CRTDSPL,DASHCIR,TRACKAC	temporary: radius of circle wanted in DASHCIR. Range in the case of other routines
RAD(I)		(3)	TATINT,TRACKAC	see input data
RADCP	TABLE(39)			see TABLE
RADDIF			TRACKAC	temporary used in navigation calculations
RADIUS			CNTRLAC	calculated radius of turn to place popup aircraft on heading to first ETA point
RAD1			TATHOLD	radius of a range marker on CRT display
RAD2			TATHOLD	same as RAD1
RAD3			TATHOLD	same as RAD1
RAD4			TATHOLD	same as RAD1
RANGE			CNTRLAC,TATOPER,TRACKAC	temporary: distance, in n mi, between two points
RATE			CNTRLAC	temporary: used for aircraft performance rate
RA20(I)	ISEQUE(1,6)-ISEQUE(3,6) SEQUE(1,6)-SEQUE(3,6)			see SEQUE
RA30(I)	ISEQUE(1,7)-ISEQUE(3,7) SEQUE(1,7)-SEQUE(3,7)			see SEQUE
RB20(I)	ISEQUE(1,8)-ISEQUE(3,8) SEQUE(1,8)-SEQUE(3,8)			see SEQUE
RB30(I)	ISEQUE(1,9)-ISEQUE(3,9) SEQUE(1,9)-SEQUE(3,9)			see SEQUE

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
RC			CNTRLAC	temporary: radius of curvature for aircraft turns in final sequencing area
RCFNM(I)		(2)	CNTRLAC	rate of climb (below 10000 ft and above 10000 ft) for aircraft being considered, in ft/n mi
RCLIMB			TRACKAC	rate of climb to reach desired altitude, in ft/scan (in sec)
RCON	TABLE (37)			see TABLE
RCP(I)		(2)	CNTRLAC,TATINT	see input data
RDESC			CNTRLAC	rate of descent (in ft/n mi) for aircraft being considered
RESET	LDISI(19)		TATMAIN	logical switch operated from real-time console to put simulation into RESET mode
RESTOR(I)		(36)	CNTRLAC	contains the current information pertinent to the resolution of conflicts along a path for a particular aircraft, using a particular combination of options available to the controller
I = 1	TETA		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	estimated time of arrival at the end of a path along which a controller is attempting to resolve any possible conflicts
2	NUMPATH		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	number of path being considered
3	OALT		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	altitude at the beginning of the path
4	AALT		CNTRLAC,CONCHK,SPDCNTR	for arrivals--assigned altitude at end of path; for departures--actual and assigned altitude at end of path
5	ETASEP		CNTRLAC,COMMUN,CONCHK,	estimated time of arrival at a separation distance from the end of a path along which the controller is attempting to resolve any possible conflicts

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
I = 6	FVEL		CNTRLAC, COMMUN, CONCHK, ETACOMP, SPDCNTR, TRACKAC	speed at the end of the path
7	DSPD		CNTRLAC, COMMUN, CONCHK, ETACOMP, SPDCNTR, TRACKAC	desired speed at the end of the path
8	TX		CNTRLAC, COMMUN, CONCHK, ETACOMP, SPDCNTR, TRACKAC	elapsed time in sec along the path before speed indicated by desired speed should be applied
9	TOTD		CNTRLAC, COMMUN, CONCHK, ETACOMP, SPDCNTR, TRACKAC	total length (n mi) of path being considered
10	NSEGM		CNTRLAC, COMMUN, CONCHK, ETACOMP, SPDCNTR, TRACKAC	number of segments that make up path being considered
11	DST2		CNTRLAC, COMMUN, CONCHK, ETACOMP, SPDCNTR, TRACKAC	distance (n mi) climbing up to 10,000 ft on path
12	DST2		CNTRLAC, COMMUN, CONCHK, ETACOMP, SPDCNTR, TRACKAC	distance (n mi) climbing above 10,000 ft on path
13	DST3		CNTRLAC, COMMUN, CONCHK, ETACOMP, SPDCNTR, TRACKAC	distance (n mi) along the path before speed indicated by desired speed should be applied
14	ISUBCK		CNTRLAC, COMMUN, CONCHK, ETACOMP, SPDCNTR, TRACKAC	subscript pointer to conflict check descriptive array (ICHKPT)
15	ISUBOP		CNTRLAC, COMMUN, CONCHK, ETACOMP, SPDCNTR, TRACKAC	subscript pointer to controller options available (IOPTN)
16	ICACT2		CNTRLAC, COMMUN, CONCHK, ETACOMP, SPDCNTR, TRACKAC	pointer to next controller action (ICNTRA) after the completion of the controller action being dealt with
17	STEMP		CNTRLAC, COMMUN, CONCHK, ETACOMP, SPDCNTR, TRACKAC	time(sec) at beginning of path determined by a previously estimated time of arrival (ETA) at that point plus a correction factor for errors at an ETA point already passed

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
I = 18	ISUBN		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	subscript of ETA array at end of path
19	ISUBO		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	subscript of ETA array at beginning of path
20	SEP		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	separation (n mi) required between aircraft at end of path
21	XLA		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	common path length (n mi) ahead of ETA at end of path
22	XLB		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	common path length (n mi) behind ETA at end of path
23	IASPCON		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	if greater than 0, pointer to an airspace array through which this aircraft wishes to fly and with which an altitude conflict check must be done
24	IDALT IOPTION(1) OPTION(1)		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	subscript (ALTOPT) of desired altitude at end of path
25	IDSPD IOPTION(2) OPTION(2)		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	subscript (IPERFS) of desired speed at end of path
26	IVECT IOPTION(3) OPTION(3)		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	pointer to path (NNPATH) being considered by the controller
27	IVECOP IOPTION(4) OPTION(4)		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	yes (1) or no(0) indicator that a compute vector option is available
28	IDVERG2 IOPTION(5) OPTION(5)		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	pointer to next divergent path available as an option for conflict resolution

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
I = 29	IALTOP IOPTION(6) OPTION(6)		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	pointer to additional altitudes available as options for conflict resolution
30	ISPDOP IOPTION(7) OPTION(7)		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	yes (1) or no (0) indicator that a speed option is available
31	ISHOP IOPTION(8) OPTION(8)		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	yes(1) or no (0) indicator that a standard hold option is available
32	IEHOP IOPTION(9) OPTION(9)		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	yes (1) or no (0) indicator that an emergency hold option is available
33	ILIMOP IOPTION(10) OPTION(10)		CNTRLAC	if greater than 0, pointer to limit aircraft array (LIMAC) that increases required separation due to specified traffic load criteria
34	OSPD		CNTRLAC,COMMUN,CONCHK, ETACOMP,SPDCNTR,TRACKAC	speed at the beginning of the path
35	TX3		CNTRLAC,ETACOMP	elapsed time along a path required to delay the desired speed change until the latest possible moment
36	INDSPD		CNTRLAC	an indicator which is set to 1 if the time to apply the desired speed change is dictated by the schedule
RLIMIT			TRACKAC	temporary: used when a range to a DME is CP mode objective
RN			TRACKAC	random number returned from RANDU
RO			ETACOMP	an additional separation distance added for common path calculation for aircraft of different speeds

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
ROLL			TATOPER,TRACKAC	bank angle in degrees of cockpit aircraft
ROLLC			TATOPER	cosine of ROLLDR
ROLLDR			TATOPER	same as ROLL, except in radians
ROLLS			TATOPER	sine of ROLLDR
ROUND			PAKSPL	round-off factor to convert reals to integers for packing
RTCL			TATOPER	cockpit rate of climb in ft/sec
RTD			CNTRLAC,CRTDSPL,INIT, TATINT,TATOPER,TRACKAC	conversion factor, radians to degrees (57.2958)
RTHETA			TRACKAC	temporary: angle used in calculating random aircraft position errors
RWYX(I,J)		(4,2)	TATINT,TATOPER	see input data
RWYY(I,J)		(4,2)	TATINT,TATOPER	see input data
RWYZ(I)		(2)	TATINT,TATOPER	see input data
RW15X(I)		(2)	TATHOLD	part of runway definition for background display of terminal area on CRT, units are in n mi
RW15Y(I)		(2)	TATHOLD	same as RW15X
RW21X(I)		(2)	TATHOLD	same as RW15X
RW21Y(I)		(2)	TATHOLD	same as RW15X
RW9LX(I)		(2)	TATHOLD	same as RW15X
RW9LY(I)		(2)	TATHOLD	same as RW15X
RW9RX(I)		(2)	TATHOLD	same as RW15X
RW9RY(I)		(2)	TATHOLD	same as RW15X
RX			CNTRLAC,TATOPER,TRACAKC	temporary: used as x component of range

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
RY			CNTRLAC,TATOPER,TRACKAC	temporary: used as y component of range
RZ			TRACKAC	temporary: used as z component of range
R1			CNTRLAC	temporary: partial distance used in sequencing logic
R15X(I)		(2)	CRTDSPL,TATHOLD	same as RW15X, except in CRT units
R15Y(I)		(2)	CRTDSPL,TATHOLD	same as R15X
R21X(I)		(2)	CRTDSPL,TATHOLD	same as R15X
R21Y(I)		(2)	CRTDSPL,TATHOLD	same as R15X
R9LX(I)		(2)	CRTDSPL,TATHOLD	same as R15X
R9LY(I)		(2)	CRTDSPL,TATHOLD	same as R15X
R9RX(I)		(2)	CRTDSPL,TATHOLD	same as R15X
R9RY(I)		(2)	CRTDSPL,TATHOLD	same as R15X

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
S	SITE(3)			see SITE
SASEP	TABLE(38)			see TABLE
SAVE(I)		(36)	CNTRLAC	temporary array to hold old RESTOR values for nominal path for original or divergent path while trying additional options
SAVE2(I)		(36)	CNTRLAC	temporary array to hold old SAVE values for the original nominal path while trying a divergent path and its options to resolve a conflict
SCAL	TABLE(13)			see TABLE
SCALE			TATINT,PAKUNI,UPKUNI	scaling factor for packing and unpacking uniform type packed words
SCLA	TABLE(40)			see TABLE
SCLE	TABLE(41)			see TABLE
SCLN	TABLE(42)			see TABLE
SECTD			TRACKAC	seconds until touchdown for aircraft under consideration on glide slope
SEP	RESTOR(20)			see RESTOR
SEPCOR			CNTRLAC	separation correction to maintain separation from ILS gate to touchdown when an aircraft is being overtaken by a faster one
SEPP(I)	TABLE(43)-TABLE(44)			see TABLE
SEPPP			CNTRLAC	temporary to be used in sequencing area for modifying SEPP(I) for the particular situation under consideration
SEQUE(I,J)	ISEQUE(I,J)	(3,16)	CNTRLAC	contains variables used in the final controller sequencing logic. Three aircraft are normally in the sequencing queue

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
J = 1	ISAC(I)	(3)	CNTRLAC	ID number of aircraft in sequencing queue
2	ISIDE(I)	(3)	CNTRLAC	hemisphere (north-1, south=2) of aircraft in sequencing queue
3	ISCLAS(I)	(3)	CNTRLAC	aircraft performance classes in sequencing queues
4	TARR(I)	(3)	CNTRLAC	time of arrival at sequencing point for aircraft in sequencing queue
5	SVEL(I)	(3)	CNTRLAC	velocity in sequencing area of aircraft in sequencing queue
6	RA20(I)	(3)	CNTRLAC	for 20° angle of intersection with final leg --distance between time of turn off of downwind leg to time of initiating turn onto localizer course
7	RA30(I)	(3)	CNTRLAC	for 30° angle of intersection with final leg --distance between time of turn off of down-wind leg to time of initiating turn onto localizer course
8	RB20(I)	(3)	CNTRLAC	for 20° angle of intersection with final leg --distance flown from time of turn-off base leg to intersection with localizer course plus distance on localizer course from a line that is the perpendicular bisector of the localizer course from the earliest turn point to the ILS gate
9	RB30(I)	(3)	CNTRLAC	for 30° angle of intersection with final leg --distance flown from time of turn-off base leg to intersection with localizer course plus distance on localizer course from a line that is the perpendicular bisector of the localizer course from the earliest turn point to the ILS gate

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
J = 10	TCP(I)	(3)	CNTRLAC	estimated time of arrival at farthest point on downwind leg of aircraft in sequencing queue
11	TOFF(I)	(3)	CNTRLAC	time to turn-off downwind leg onto base leg for aircraft in sequencing queue
12	TONN(I)	(3)	CNTRLAC	time to turn from base leg to final leg for aircraft in sequencing queue
13	TGATE(I)	(3)	CNTRLAC	estimated time at ILS gate for aircraft in sequencing queue
14	ANG(I)	(3)	CNTRLAC	the angle of intersection of the base leg with the final leg of the sequencing pattern for an aircraft in the sequencing queue
15	DL(I)	(3)	CNTRLAC	distance between sequencing point and earliest turn-off point
16	FETA(I)	(3)	CNTRLAC	subscript (ISUBN) of ETA at terminal end of sequencing area path
SHOLDD			CNTRLAC,TATINT	see input data
SI			XFORM	sine of PSIREL
SIAR			CRTDSPL,XFORM	sine of (PSIREL - PSI)
SIG(I)		(3)	TATINT,TRACKAC	see input data
SITE(I)		(3)	TRACKAC	temporary for particular navaid site under consideration
I = 1	CPX		TRACKAC	north-south coordinate of navaid site
2	CPY		TRACKAC	east-west coordinate of navaid site
3	S		TRACKAC	standard deviation associated with a navaid site used to apply flight path errors to an aircraft using the navaid
SPD			TRACKAC	aircraft speed used to update position. It is either from last scan or new value computed this scan

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
SPDF			TRACKAC	future speed of aircraft being position updated
SPDSAV			CNTRLAC	when trying speed deviations to resolve a conflict, SPDSAV is the original speed used
SPDSTD(I,J)		(9,2)	TATINT, CNTRLAC	input data--the speed deviation to use on either side of the optimal speed
SPEED			CNTRLAC	temporary used for speed from performance table
SQSCALE	TABLE(46)			see TABLE
SQX(I)		(6)	TATINT,CRTDSPL,TATHOLD	see input data
SQY(I)		(6)	TATINT, CRTDSPL,TATHOLD	see input data
STEMP	RESTOR(17)			see RESTOR
SVEL	ISEQUE(1,5)-ISEQUE(3,5) SEQUE(1,5)-SEQUE(3,5)			see SEQUE
SWITCH(I)		(10)	CNTRLAC,TATREST	an array of on-off switches used for flow control; one is assigned for each traffic situation desired

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
T(I)		(35)	CNTRLAC,COMMUN,TRACKAC	for each aircraft in active traffic, T(I) is time aircraft has been on present path segment
TA			COMETA	time to accelerate/decelerate from initial speed (VI) to final speed (VF)
TAATMFL			TATMAIN	literal argument, i.e. "TAATMFL," in call list to a system routine
TABLE(I)		(75)	TATINT	a table of real variables in core storage that are easily displayed individually on the real-time console. Used to look up or display variables
I = 1	AK		TATINT,TRACKAC	see input data
2	ALPHK		TATINT,TRACKAC	see input data
3 thru 6				not currently used
7	XREL		TATINT,TRACKAC,XFORM	x,y,z coordinates in terminal area of aircraft being used as coordinate origin for the relative position of all other aircraft to him
8	YREL		TATINT,TRACKAC,XFORM	
9	ZREL		TATINT,TRACKAC	
10	PSIREL		CRTDSPL,TATINT,TRACKAC, XFORM	heading used in relative display as reference to which all other terminal display is relative
11	TEMPS		CNTRLAC,COMMUN,TATREST, TATHOLD,TATOPER,TRACKAC	temporary
12	VECERR		TATINT,CNTRLAC	see input data
13	CRTSCAL,SCAL		CRTDSPL,INIT,XFORM	scale factor for CRT display which converts n mi to CRT units
14	DELALT		TATINT,TRACKAC	see input data

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
TABLE(I)				
I = 15	DELSPD		TATINT,TRACKAC	see input data
16	DELT		CNTRLAC,TATINT,TATREAD, TRACKAC	see input data
17				not currently used
18				not currently used
19	DKCON		TATINT,TRACKAC	see input data
20	FREQ		INIT,TATHOLD	desired CRT update time (frequency)
21	H		INIT	the time interval in real-time seconds/cycle
22				not currently used
23	PASEP		CNTRLAC,TATINT	see input data
24	PDSEP		CNTRLAC,TATINT	see input data
25	PHIK		TATINT,TRACKAC	see input data
26	BETA		TATINT,TATREST	see input data
27				not currently used
28				not currently used
29	UPERSC		INIT,TATOPER	scaling factor for u coordinate for perspective plot of runway
30	VPERSC		INIT,TATOPER	scaling factor for v coordinate for perspective plot of runway
31 32 33 34 35	DELX1 DELX2 DELX3 DELY1 DELY2		INIT,CRTDSPL	increments in x and y for beginning points of various components for the tag
36				not currently used

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
TABLE(I)				
I = 37	RCON		TATINT,TRACKAC	see input data
38	SASEP		CNTRLAC,TATINT	see input data
39	RADCP		INIT,TRACKAC	radius in n mi for "making good" an objective
40				not currently used
41				not currently used
42				not currently used
43	SEPP(1)		TATINT,CNTRLAC	see input data
44	SEPP(2)		TATINT,CNTRLAC	see input data
45				not currently used
46	SQSCALE		INIT,TATHOLD	scale factor for soze of box symbol for popup aircraft
47	VELCON		INIT,TATOPER	scaling factor for cockpit displays
48	VECSD		TATINT,TRACKAC	see input data
49 thru 52				not currently used
53	CKROLL		TATINT,TATOPER	see input data
54	CKRC		TATINT,TATOPER	see input data
55	CROLL		TATINT,TATOPER	see input data
56	CRC		TATINT,TATOPER	see input data
57	FDELT		TATINT,TATOPER	time between real-time frames. Used for updating cockpit aircraft position each frame
58	ZCON		TATINT,TATOPER	see input data

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
TABLE(I)				
I = 59	HDCON		TATINT,TATOPER	see input data
60 thru 72	IPOPUP(I) POPUP(I)	(13) (13)	TRACKAC	an array of data, input from the real-time console, describing initial conditions of a popup aircraft
73 74 75	XSTP YSTP VSTP		INIT,TATOPER	scaling factors for cockpit displays of x, y, position, and velocity
TAFIX			CNTRLAC	time of arrival at sequencing point for aircraft under consideration
TANG			CNTRLAC	change in current heading to obtain desired heading for popup aircraft
TARR(I)	ISEQUE(1,4)-ISEQUE(3,4) SEQUE(1,4)-SEQUE(3,4)	(3)	CNTRLAC	see SEQUE
TATHOLD			TATMAIN	literal, i.e., "TATHOLD," used in call list to system routine
TATOPER			TATMAIN	literal, i.e., "TATOPER," used in call list to system routine
TATREST			TATMAIN	literal, i.e., "TATREST," used in call list to system routine
TCMSG(I)	OUT(3)-OUT(8)	(6)	COMMUN	temporary array into which the ICMSG2 of interest is placed
TCMSG2(I)	OUT(9)-OUT(11)	(3)	COMMUN	temporary array into which the ICMSG3 of interest is placed
TCP(I)	ISEQUE(1,10)-ISEQUE(3,10) SEQUE(1,10)-SEQUE(3,10)			see SEQUE

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
TEMETA			CNTRLAC	temporary ETA time
TEMGAT			CNTRLAC	temporary estimate of time at gate for aircraft being sequenced
TEMOFF			CNTRLAC	temporary estimate of time to turn-off downwind path for aircraft being sequenced
TEMONN			CNTRLAC	temporary estimate of time to turn onto the final path for aircraft being sequenced
TEMP			CNTRLAC,COMMUN,CONCHK, TATOPER,SPDCNTR,TRACKAC	temporary
TEMPA(I)	ITEMPA(I)	(10)	CNTRLAC,CONCHK,CRTDSPL, TATOPER,SPDCNTR,TRACKAC, COMMUN	temporary array
TEMPA3(I)		(10)	CNTRLAC	temporary array
TEMPS	TABLE(11)			see TABLE
TEMPS2			COMMUN	temporary
TEMP1			ETACOMP	temporary for elapsed time required to fly a distance with speed change occurring first
TEMP2			ETACOMP	temporary, same as TEMP1 except with speed change occurring at latest possible time
TEMSEP			CNTRLAC	temporary used as estimated time of arrival at separation distance from ETA point
TERM	LDISI(20)		TATMAIN	console switch to take real-time simulation to terminate mode
TETA	RESTOR(1)			see RESTOR
TETAE			CNTRLAC,ETACOMP,SPDCNTR	earliest time of arrival at a point for aircraft under consideration that can result by changing time of applying speed change
TETAL			CNTRLAC,ETACOMP,SPDCNTR	same as TETAE but latest time of arrival

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
TETAO			CNTRLAC,ETACOMP	temporary ETA time at beginning ETA point on path being considered
TETAOM			CNTRLAC	temporary for the latest ETA time at the beginning of a path of those ETA times for aircraft that need priority over an aircraft that is checking for conflicts along the path and needs a 360° hold
TGATE(I)	ISEQUE(1,13)-ISEQUE(3,13) SEQUE(1,13)-SEQUE(3,13)			see SEQUE
TH			DASHCIR	used to break circumference of circle into segments for dashed lines
THETA			CNTRLAC	temporary used for difference between aircraft heading and desired heading
TIME			CNTRLAC,COMMUN,TATOPER	time, in seconds, from start of simulation
TLE(I)		(40)	TATHOLD	an array of literals used for labeling the display on the CRT, e.g., date, run number, etc.
TMSGA(I)		(6)	CNTRLAC	temporary array for elements to pack controller message
I = 1	XDENT		CNTRLAC	same as element 1 of IFCMSG2
2	XMSGL		CNTRLAC	same as element 2 of IFCMSG2
3	XLTT		CNTRLAC	same as element 3 of IFCMSG2
4	XCLEAR		CNTRLAC	same as element 4 of IFCMSG2
5	XITYPE		CNTRLAC	same as element 5 of IFCMSG2
6	XETA		CNTRLAC	same as element 6 of IFCMSG2

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
TMSGA2(I)		(3)	CNTRLAC	temporary array for elements to pack controller message
I = 1	XMINFO1		CNTRLAC	same as element 1 of IFCMSG3
2	XMINFO2		CNTRLAC	same as element 2 of IFCMSG3
3	XMINFO3		CNTRLAC	same as element 3 of IFCMSG3
TOFF(I)	ISEQUE(1,11)-ISEQUE(3,11) SEQUE(1,11)-SEQUE(3,11)			see SEQUE
TONN(I)	ISEQUE(1,12)-ISEQUE(3,12) SEQUE(1,12)-SEQUE(3,12)			see SEQUE
TOTD	RESTOR(9)			see RESTOR
TOTD2			ETACOMP	temporary used as the distance along a path up to the separation distance from the point at terminal end of the path
TR			CNTRLAC	temporary used as turn rate
TRIX(I)		(5)	CRTDSPL,TATINT	see input data
TRIY(I)		(5)	CRTDSPL,TATIN	see input data
TRYETA			SPDCNTR	a temporary ETA used while developing time to change speed to resolve a conflict
TSKED(I)		(7)	TRACKAC	a temporary array used when packing/unpacking active schedule
I = 1	XMODE		TRACKAC	flight mode (see IASKED packed word description for more complete definitions of these 7 variables)
2	XINFO1		TRACKAC	INFO 1

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
TSKED(I)				
I = 3	XINFO2		TRACKAC	INFO 2
4	XFALT		TRACKAC	altitude scheduled
5	XMODECP		TRACKAC	MODECP--mode to obtain objective
6	XINFO3		TRACKAC	INFO 3
7	XINFO4		TRACKAC	INFO 4
TTH			CNTRLAC	temporary used as time to begin hold
TURN			CNTRLAC	temporary for type of turn--left or right
TVI			COMETA,ETACOMP	computed actual speed after flying a distance
TX	RESTOR(8)			see RESTOR
TXA			CNTRLAC	temporary XA--coordinate of inner fix
TXC			CNTRLAC	temporary XC--coordinate of final sequencing point
TX1			CNTRLAC	temporary aircraft x coordinate while being sequenced
TX2			ETACOMP	time in seconds that aircraft will not be climbing or changing speed on path being considered
TX3	RESTOR(35)			see RESTOR
TYA			CNTRLAC	temporary YA coordinate of inner fix
TYBUF(I)		(16)	CNTRLAC,COMMUN,TRACKAC	buffer used for typewritten messages
TYC			CNTRLAC	temporary YC--coordinate of final sequencing point
TYI			CNTRLAC	temporary aircraft y coordinate while being sequenced

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
UPERS(I)		(4)	TATOPER	u coordinates for perspective runway display on CRT
UPERSC	TABLE(29)			see TABLE
V			CRTDSPL,TATOPER	temporary used for aircraft speed
VARCHNG			TATINT	logical variable used as a flag in system routine call list
VC			TRACKAC	temporary--closing velocity
VECERR	TABLE(12)			see TABLE
VECSD	TABLE(48)			see TABLE
VEL			TATOPER	velocity scaled for cockpit aircraft
VELCON	TABLE(47)			see TABLE
VELFT			TATOPER	velocity in ft/sec
VF			COMETA,ETACOMP	final velocity on path being considered
VFT			TRACKAC	velocity in ft/sec
VI			COMETA,ETACOMP	initial velocity on path being considered
VPERS(I)		(4)	TATOPER	v coordinates for perspective runway display on CRT
VPERSC	TABLE(30)			see TABLE
VSTP	TABLE(75)			see TABLE
VX			TRACKAC	x component of velocity

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
VY			TRACKAC	y component of velocity
V1			CONCHK,ETACOMP,SPDCNTR	temporary for velocity of first of two aircraft being considered on a path
V2			CONCHK,ETACOMP,SPDCNTR	temporary for velocity of second of two aircraft being considered on a path
WRTTPE	LOGIC(7)			see LOGIC

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
X		dimensions vary according to routine in which used; max. = 15	DASHCIR, TATHOLD, XFORM	temporary for north-south coordinates
XA(I)		(2)	CNTRLAC, TATINT	see input data
XB			CNTRLAC	temporary for x coordinate of earliest turn-off point on downwind leg for an aircraft
XBOX	TABLE(49)			see TABLE
XC(I)		(2)	CNTRLAC, TATINT	see input data
XCLEAR	TMSGA(4)			see TMSGA
XCLR2			CNTRLAC	temporary--number of path segments for which to issue controller clearances
XCP	CPCOORD(1)			see CPCOORD
XDENT	TMSGA(1)			see TMSGA
XDILS(I)		(2)	CNTRLAC, TATINT, TATOPER	see input data
XDIST(I)		(2)	CNTRLAC, TATINT	see input data
XETA	TMSGA(6)			see TMSGA
XFINAL(I)		(2)	CNTRLAC, TATINT	see input data
XI			CNTRLAC, CRTDSPL	temporary--north-south coordinate
XINFO1	TSKED(2)			see TSKED
XINFO2	TSKED(3)			see TSKED
XINFO3	TSKED(6)			see TSKED
XINFO4	TSKED(7)			see TSKED

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
XIT			CNTRLAC,TATREST,PAKSPL	in PAKSPL/UPKSPL it is the array or scalar unpacked word(s). In other routines it is the floating version of IT (scan counter)
XITYPE	TMSGA(5)			see TMSGA
XKALP			TRACKAC	temporary used for an angle
XKALPT			TRACKAC	temporary used for an angle
XKNFS			INIT	conversion factor, knots to ft/sec
XLA	RESTOR(21)			see RESTOR
XLB	RESTOR(22)			see RESTOR
XLID			CNTRLAC	computed length of base leg in sequencing area for aircraft under consideration
XLIND(I)		(7)	CNTRLAC	unpacked elements of LIMAC--see packed word description for LIMAC
XLTT	TMSGA(3)			see TMSGA
XLTT2			CNTRLAC	a temporary for latest transmit time
XMINFO1	TMSGA2(1)			see TMSGA2
XMINFO2	TMSGA2(2)			see TMSGA2
XMINFO3	TMSGA2(3)			see TMSGA2
XMSGL	TMSGA(2)			see TMSGA
XMSGL2			CNTRLAC	temporary used for message length
XMTAG			CRTDSPL	used as test for whether a tag can be displayed on the CRT for an aircraft without overwriting another aircraft
XN			TRACKAC	coordinate of aircraft being considered
XNFRAM			TATINT	same as the integer NFRAM but in floating point form

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
XNREF(I)		(4)	TATINT,CRTDSPL	x coordinates for true north arrow on CRT for relative display
XOWN			CRTDSPL,TATOPER,TRACKAC	part of x,y,z coordinates of cockpit aircraft
XPL(I)		(35)	CNTRLAC,TRACKAC	x coordinate at beginning of path segment Ith aircraft is currently flying
XPTAG			CRTDSPL	same as XMTAG
XREL	TABLE(7)			see TABLE
XRET			TRACKAC,UPKUNI	temporary used for returning packed variable from UPKUNI through call list
XSTP	TABLE(73)			see TABLE
XT1			CRTDSPL	temporary--one of the coordinates used for developing the CRT tag on an aircraft
XT2			CRTDSPL	same as XT1
XT3			CRTDSPL	same as XT1
XWORK(I)		(150) (25)	TATINT TATREAD	temporary arrays used for working space while reading and packing data
XX			CRTDSPL,XFORM	temporary for x coordinate in relative CRT display

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
Y		dimensions vary according to routine in which used; max. = 15	DASHCIR,XFORM, TATHOLD	temporary for east-west coordinates
YA(I)		(2)	CNTRLAC,TATINT	see input data
YAWC			TATOPER	cosine of heading of cockpit aircraft
YAWS			TATOPER	sine of heading of cockpit aircraft
YB			CNTRLAC	temporary for y coordinate of earliest turn-off point on downwind leg for an aircraft
YBOX	TABLE(50)			see TABLE
YC(I)		(2)	CNTRLAC,TATINT	see input data
YCP	CPCOORD(2)			see CPCOORD
YFL			RANDU	the random number generated by the routine passed through call list to calling program
YI			CNTRLAC,CRTDSPL	temporary--east-west coordinate
YN			TRACKAC	same as XN
YNREF(I)		(4)	TATINT,CRTDSPL	same as XNREF except y coordinates
YOWN			CRTDSPL,TATOPER,TRACKAC	same as XOWN
YPL(I)		(35)	CNTRLAC,TRACKAC	same as XPL(I) but y coordinate
YPTAG			CRTDSPL	same as XMTAG
YREL	TABLE(8)			see TABLE
YS			CRTDSPL,TATHOLD	starting y coordinate used for writing controller messages on CRT
YSTP	TABLE(74)			see TABLE
YT1			CRTDSPL	same as XT1

Table E-1. Continued.

Name	Aliases	Dimensions	Routines in Which Variable is Used	Definition
YT2			CRTDSPL	same as XT1
YWORK(I)		(10)	TATINT,TATREAD	temporary working array used in reading in and packing data
YY			CRTDSPL,XFORM	same as XX but y coordinate
Y1			TRACKAC	a random number between 0 and 1
Y2			TRACKAC	variable used for summoning Y1 values to obtain normalized distribution of random numbers
ZCON	TABLE(58)			see TABLE
ZN			TRACKAC	same as XN
ZOWN			TATINT,TATOPER,TRACKAC	same as XOWN
ZREL	TABLE(9)			see TABLE

APPENDIX F

Description of Packed Computer Words

In order to minimize core storage requirements where feasible, several elements of data are packed into each computer word. This appendix defines these packed words and the data elements contained in each packed word. Because only integers may be effectively packed, attribute tables define scaling factors, number of bits required, the element position in the packed word, and the algebraic sign of the element. The packed computer words are:

Packed Array	Attribute Table	Page No.
IACFPER - aircraft error	-	247
IASKED - active schedule of all current and future path segments for each aircraft in active traffic	IPASKED	248
IATAS - actual time of arrival	IPTAS	249
IATRAF21 - current or future path information	IPAPATH	250
IATRAF23 - current or future path information	IPAPATH	250
ICHKPT - conflict check descriptive array	IPCHKPT	251
ICMSG1 - current controller message	-	252
ICMGS2 - current controller message	IPMSG2	252
ICMSG3 - current controller message	IPMSG3	253
ICNTRA1 - controller action point	IPCNTR1	255
ICNTRA2 - controller action point	IPCNTR2	256
IDROUTE - divergent path (route) description	IPDRT	257
IETA1 - estimated time of arrival array	IPETA1	258
IETA2 - estimated time of arrival array	IPETA2	259
IETAS - estimated time of arrival	IPTAS	260
IETCOOR - ETA location coordinates	IPETC	261
IFCMSG1 - future controller message	-	262
IFCMSG2 - future controller message	IPMSG2	262
IFCMSG3 - future controller message	IPMSG3	263
INAV - navigational aids (ground-based)	IPNSITE	264

Aircraft Error. IACFPER

Each aircraft performance class has associated with it a standard deviation of heading error parameter.

Dimension: 3

Packed words for each entry: 1/10

Aliases: none

IACFPER

(packed 10 elements in each computer word)

Characteristics of Each Element

Description: performance parameter, standard deviation of heading error

Form: X.X

Max. Value: 63

Scale Factor: 10^{-1}

Binary Bits: 6

Packed Computer Word Diagram

Element	1	2	3	4	5	6	7	8	9	10
Bits	6	6	6	6	6	6	6	6	6	6

Uniform Pack/Unpack routines used; no attribute tables required.

Active Schedule of all Current and Future Path Segments for Each Aircraft in Active Traffic. IASKED

Describes the four-dimensional flight plan using both aircraft and ground-based navigational techniques.

Dimension: 35 by 11

Packed words for each entry: 1

Aliases: none

IASKED

Element Number	Description*	Form	Max. Value	Scale Factor	Binary Bits
1.	Flight navigational mode (same as ISKED element)	XX	15	10^0	4
2.	Information No. 1	XXX	511	10^0	9
3.	Information No. 2	XXX	511	10^0	10
4.	Altitude in hundreds of feet	XXX	511	10^{-2}	9
5.	Control point mode, i.e., type of make good conditions (same as ISKED element 6)	XX	15	10^0	4
6.	Information No. 3	XXX	511	10^0	9
7.	Information No. 4	XXXX.	± 16383	10^0	15
		Total Bits Used (no excess)			60

*See tables associated with ISKED for details.

Packed Computer Word Diagram

Element	1	2	3	4	5	6	7
Bits	4	9	10	9	4	9	15

Attribute table IPASKED (dimensioned 4 by 7) describes the packed word IASKED.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4	5	6	7
Length	4	9	10	9	4	9	15
Sign	0	0	1	0	0	0	1
Scale	0	0	0	-2	0	0	0
First Shift	56	47	37	28	24	15	0

Actual Time of Arrival. IATAS

Describes actual time of arrival information for an aircraft at each ETA point on its route.

Dimension: 35 by 2

Packed words for each entry: 1/4

Aliases: none

IATAS

(packed 4 elements in each computer word)

Characteristics of Each Element

Description: Actual time of arrival at each ETA for an aircraft to be used for post-processing by comparing with estimated time of arrival. Also used as correction factor to keep ETA errors from growing along the route

Form: XXXX (4-sec scans)

Max. Value: 32767

Scale Factor: 10°

Binary Bits: 15

Packed Computer Word Diagram

Element	1	2	3	4
Bits	15	15	15	15

Attribute table IPTAS (dimensioned 4 by 4) describes the packed word IATAS.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4
Length	15	15	15	15
Sign	0	0	0	0
Scale	0	0	0	0
First Shift	45	30	15	0

Current (IATRF21) or Future (IATRF23) Path Information.

IATRF21 describes current path in active traffic array.

IATRF23 describes future path in active traffic array.

Both have the same element descriptions.

Dimension: 35

Packed words for each entry: 1

Aliases: none

IATRF21
IATRF23

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	Altitude (thousands of ft) at beginning end of the path	XX	63	10^3	6
2.	Altitude (thousands of ft) at terminal end of the path	XX	63	10^3	6
3.	Path number NNPATH	XXX	127	10°	7
4.	Subscript of ETA array at beinning end of the path	XXX	127	10°	7
5.	Subscript of ETA array at terminal end of the path	XXX	127	10°	7
		Total Bits Used (27 excess)			33

Packed Computer Word Diagram

Element	Excess	1	2	3	4	5
Bits	27	6	6	7	7	7

Attribute table IPAPATH (dimensioned 4 by 5) describes the packed words IATRF21 and IATRF23.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4	5
Length	6	6	7	7	7
Sign	0	0	0	0	0
Scale	-3	-3	0	0	0
First Shift	27	21	14	7	0

Conflict Check Descriptive Array. ICHKPT

Describes details of the path where conflict checking is to be done.

Dimension: 40

Packed words for each entry: 1

Aliases: none

ICHKPT

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	Subscript of ETA array (IETA) at terminal end of path	XXX.	127	10^0	7
2.	Subscript of ETA array (IETA) at beginning end of path	XXX.	127	10^0	7
3.	Separation (n mi)	XX.X	127	10^0	7
4.	Common path length (XLA in n mi) ahead of ETA	XX.X	511	10^{-1}	9
5.	Common path length (XLB in n mi) behind ETA	XX.X	511	10^{-1}	9
6.	Pointer to airspace conflict array (ASPACE)	XX	63	10^0	6
		Total Bits Used (15 excess)			45

Packed Computer Word Diagram

Element	Excess	1	2	3	4	5	6
Bits	15	7	7	7	9	9	6

Attribute table IPCHKPT (dimensioned 4 by 6) describes the packed word ICHKPT.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4	5	6
Length	7	7	7	9	9	6
Sign	0	0	0	0	0	0
Scale	0	0	1	1	1	0
First Shift	38	31	24	15	6	0

Current Controller Messages. ICMSG1, ICMSG2, ICMSG3

Messages ready to be transmitted immediately are in this array for each controller.

Dimensions: 8 by 5 per variable

Packed words for each message: 2
(also 1 non-packed word)

Aliases: none

ICMSG1

Non-packed, latest transmit time in 4-second scans.

ICMSG2

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	Aircraft ID	XXX	1023	10^0	10
2.	Total transmit time in 4-second scans, i.e., message length	XX	15	10^0	4
3.	Earliest transmit time in 4-second scans	XXXX	32767	10^0	15
4.	Number of segment clearances being issued	XX	15	10^0	4
5.	Message type	XX	31	10^0	5
6.	Subscript of ETA array associated with terminal end of path to which message applies	XXX	127	10^0	7
		Total Bits Used (15 excess)			45

Packed Computer Word Diagram

Element	Excess	1	2	3	4	5	6
Bits	15	10	4	15	4	5	7

Attribute table IPMSG2 (dimensioned 4 by 6) describes the packed word ICMSG2.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4	5	6
Length	10	4	15	4	5	7
Sign	0	0	0	0	0	0
Scale	0	0	0	0	0	0
First Shift	35	31	16	12	7	0

ICMSG3

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	Information No. 1 (altitude in hundreds of feet, or speed in knots, or ± heading in degrees, or intersection ID, i.e., outer fix) (altitude must be scaled in calling routine)	±XXX	±511	10^0	13
2.	Information No. 2 (altitude in hundreds of feet, or speed in knots) (altitude must be scaled in calling routine)	XXX	511	10^0	9
3.	Information No. 3 (altitude in hundreds of feet)	XXX	511	10^2	9
		Total Bits Used (29 excess)			31

Packed Computer Word Diagram

Element	Excess	1	2	3
Bits	29	13	9	9

Attribute table IPMSG3 (dimensioned 4 by 3) describes the packed word ICMSG3.

Attribute Table for Pack/Unpack Routines

Element	1	2	3
Length	13	9	9
Sign	1	0	0
Scale	0	0	-2
First Shift	18	9	0

Controller Action Points. ICNTRA1, ICNTRA2

Describes each place where controller action is desired. Types of controller action included in the program are:

1. Arrival
2. Departure
3. General Conflict
4. Sequencing
5. No-conflict-check, but compute ETA
6. Pop-up
7. Re-check action on present path
8. Hand-off to another controller
9. Register aircraft in airspace conflict array (ASPACE)
10. Remove aircraft from airspace conflict array (ASPACE)
11. Compute new vector to checkpoint.

Dimensions: 85 by 2 per controller action point

Packed words for each entry: 2

Aliases: none

ICNTRA1(i)

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	Type of controller action to be done	XX.	31	10°	5
2.	x coordinate (n mi North)	±XX.XX	±8191	10^{-2}	14
3.	y coordinate (n mi East)	±XX.XX	±8191	10^{-2}	14
4.	Pointer to ICHKPT (conflict check) array	XXX.	127	10°	7
5.	Pointer to IOPTN (option) array or to ASPACE (air-space) array if controller action is type 9 or 10	XXX.	127	10°	7
6.	Pointer to next controller action subscript	XXX.	127	10°	7
7.	Nominal heading	XX.	63	10°	6
		Total Bits Used (No excess)			60

Packed Computer Word Diagram

Element	1	2	3	4	5	6	7
Bits	5	14	14	7	7	7	6

The attribute table IPCNTR1 (dimensioned 4 by 7) describes the packed word ICNTRA1.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4	5	6	7
Length	5	14	14	7	7	7	6
Sign	0	1	1	0	0	0	0
Scale	0	2	2	0	0	0	0
First Shift	55	41	27	20	13	6	0

ICNTRA2(i)

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	Controller number responsible for this action	XX	15	10°	4
2.	Distance (n mi) to ending checkpoint from beginning checkpoint	XX.X	511	10^{-1}	9
		Total bits used (47 excess)			13

Packed Computer Word Diagram

Element	Excess	1	2
Bits	47	4	9

Attribute table IPCNTR2 (dimensioned 4 by 2) describes the packed word ICNTRA2.

Attribute Table for Pack/Unpack Routines

Element	1	2
Length	4	9
Sign	0	0
Scale	0	1
First Shift	9	0

Divergent Path Description. IDROUTE(i)

Describes which divergent path to use if option is available. (Value of "i" is from 5th element of IOPTN.)

Dimension: 0

Packed words for each entry: 1

Aliases: none

IDROUTE Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	Subscript for ICHKPT array	XXX	127	10^0	7
2.	Subscript for IOPTN array	XXX	127	10^0	7
3.	Subscript for ICNTRA array	XXX	127	10^0	7
4.	Subscript for another divergent path in this array if available	XXX	127	10^0	7
5.	Distance (n mi) between ends of this path, i.e., path length	XXX	511	10^{-1}	9
		Total Bits Used (23 excess)			37

Packed Computer Word Diagram

Element	Excess	1	2	3	4	5
Bits	23	7	7	7	7	9

Attribute table IPDRT (dimensioned 4 by 5) describes the packed word IDROUTE.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4	5
Length	7	7	7	7	9
Sign	0	0	0	0	0
Scale	0	0	0	0	1
First Shift	30	23	16	9	0

Estimated Time of Arrival Array. IETA1, IETA2

Describes information associated with all aircraft scheduled to arrive at a point or points associated with this array.

Dimensions: 36 by 2 per variable

Packed Words for Each Entry: 2

Aliases: none

IETA1

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	Aircraft ID number	XXXX	1023	10°	10
2.	Time (in sec) estimated for arrival over a point associated with this ETA array	XXXXXX.XX	41943.04	10^{-2}	22
3.	Speed (in n mi/sec) over the point associated with this ETA array	.XXXXXXX	.2097152	10^{-7}	21
4.	Controller action number which placed aircraft in this array	XXX	127	10°	7
		Total Bits Used (no excess)			60

Packed Computer Word Diagram

Element	1	2	3	4
Bits	10	22	21	7

Attribute table IPETA1 (dimensioned 4 by 4) describes the packed word IETA1.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4
Length	10	22	21	7
Sign	0	0	0	0
Scale	0	2	7	0
First Shift	50	28	7	0

IETA2

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	Subscript of ETA array at beginning end of path	XXX	127	10^0	7
2.	Subscript of ETA array at terminal end of path	XXX	127	10^0	7
3.	Path number	XXX	127	10^0	7
4.	Assigned altitude (integer thousands of feet) at beginning end of path	XX	63	10^3	6
5.	Assigned altitude (integer thousands of feet) at terminal end of path	XX	63	10^3	6
6.	ETA (in sec) at separation distance from ETA location	XXXXXX.XX	167772.15	10^{-2}	24
		Total Bits Used (3 excess)			57

Packed Computer Word Diagram

Element	Excess	1	2	3	4	5	6
Bits	3	7	7	7	6	6	24

Attribute table IPETA2 (dimensioned 4 by 6) describes the packed word IETA2.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4	5	6
Length	7	7	7	6	6	24
Sign	0	0	0	0	0	0
Scale	0	0	0	-3	-3	2
First Shift	50	43	36	30	24	0

Estimated Time of Arrival. IETAS

Describes estimated time of arrival information for an aircraft at each ETA point on its route.

Dimensions: 35 by 2

Packed words for each entry: 1/4

Aliases: none

IETAS

(packed 4 elements in each computer word)

Characteristics of Each Element

Description: Estimated time of arrival at each ETA for an aircraft to be used for post-processing by comparing with actual time of arrival

Form: XXXX (scans)

Max. Value: 32767

Scale Factor: 10^{0}

Binary Bits: 15

Packed Computer Word Diagram

Element	1	2	3	4
Bits	15	15	15	15

Attribute table IPTAS (dimensioned 4 by 4) describes the packed word IETAS.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4
Length	15	15	15	15
Sign	0	0	0	0
Scale	0	0	0	0
First Shift	45	30	15	0

ETA Location Coordinates. IETCOOR

Describes location of ETA and stores information on whether any aircraft are in a standard holding pattern at the location.

Dimensions: 36

Packed words for each entry: 1

Aliases: none

IETCOOR

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	x coord. (n mi north)	±XX.XX	±8191	10^{-2}	14
2.	y coord. (n mi east)	±XX.XX	±8191	10^{-2}	14
3.	Binary switch (1=yes) to indicate there is a propeller aircraft in a standard hold pattern on path ending with this ETA point	X	1	10^{0}	1
4.	Same as 3 except indicates jet aircraft	X	1	10^{0}	1
		Total Bits Used (30 excess)			30

Packed Computer Word Diagram

Element	Excess	1	2	3	4
Bits	30	14	14	1	1

Attribute table IPETC (dimensioned 4 by 4) describes the packed word IETCOOR.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4
Length	14	14	1	1
Sign	1	1	0	0
Scale	2	2	0	0
First Shift	16	2	1	0

Future Controller Messages. IFCMSG1, IFCMSG2, IFCMSG3

Messages to be transmitted later are in this array for each controller.

Dimensions: 8 by 25 per variable

Packed words for each message: 2
(also 1 non-packed word)

Aliases: none

IFCMSG1

Non-packed, earliest transmit time in 4-second scans.

IFCMSG2

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	Aircraft ID	XXX	1023	10^0	10
2.	Total transmit time in 4-second scans, i.e. message length	XX	15	10^0	4
3.	Latest transmit time in 4-second scans	XXXX	32767	10^0	15
4.	Number of segment clearances being issued	XX	15	10^0	4
5.	Message type	XX	31	10^0	5
6.	Subscript of ETA array associated with terminal end of path to which message applies	XXX	127	10^0	7
		Total Bits Used (15 excess)			45

Packed Computer Word Diagram

Element	Excess	1	2	3	4	5	6
Bits	15	10	4	15	4	5	7

Attribute table IPMSG2 (dimensioned 4 by 6) describes the packed word IFCMSG2.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4	5	6
Length	10	4	15	4	5	7
Sign	0	0	0	0	0	0
Scale	0	0	0	0	0	0
First Shift	35	31	16	12	7	0

IFCMSG3

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	Information No. 1 (altitude in hundreds of feet, or speed in knots, or ±heading in degrees, or intersection ID, i.e. outer fix) (altitude must be scaled in calling routine)	±XXX	±511	10^0	13
2.	Information No. 2 (altitude in hundreds of feet, or speed in knots) (altitude must be scaled in calling routine)	XXX	511	10^0	9
3.	Information No. 3 (altitude in hundreds of feet)	XXX	511	10^2	9
		Total Bits Used (29 excess)			31

Packed Computer Word Diagram

Element	Excess	1	2	3
Bits	29	13	9	9

Attribute table IPMSG3 (dimensioned 4 by 3) describes the packed word IFCMSG3.

Attribute Table for Pack/Unpack Routines

Element	1	2	3
Length	13	9	9
Sign	1	0	0
Scale	0	0	-2
First Shift	18	9	0

Navigational Aids (Ground-Based). INAV

Describes each navigational aid site.

Dimensions: 10 by 5

Packed words for each entry: 1

Aliases: INAV(i,1) or IADF(i)
INAV(i,2) or IVOR(i)
INAV(i,3) or IFIX(i)
INAV(i,4) or IDME(i)
INAV(i,5) or IXILS(i)

INAV

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	x coord. (n mi north)	±XX.XX	±8191	10^{-2}	14
2.	y coord. (n mi east)	±XX.XX	±8191	10^{-2}	14
3.	Standard deviation (error)	X.X	127	10^{-1}	7
4.	"Make good limit" (n mi)	X.X	63	10^{-1}	6
		Total Bits Used (19 excess)			41

Packed Computer Word Diagram

Element	Excess	1	2	3	4
Bits	19	14	14	7	6

Attribute table IPNSITE (dimensioned 4 by 4) describes the packed word INAV.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4
Length	14	14	7	6
Sign	1	1	0	0
Scale	2	2	1	1
First Shift	27	13	6	0

Controller Options Available. IOPTN

Describes which options are available for controller action points.

Dimension: 55

Packed words for each entry: 1

Aliases: none

IOPTN

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	Subscript of desired altitude (ALTOPT)	XXX	127	10°	7
2.	Subscript of desired speed (IPERFS)	XXX	127	10°	7
3.	Subscript of nominal path (NNPATH)	XXX	127	10°	7
4.	Computed vector option (not currently used in model)	XX	15	10°	4
5.	Subscript of divergent path (IDROUTE)	XX	63	10°	6
6.	Altitude option availability (yes-no)	XX	63	10°	6
7.	Speed change availability (yes-no)	XX	63	10°	6
8.	Standard hold option availability (yes-no)	X	1	10°	1
9.	Emergency hold option availability (yes-no)	X	1	10°	1
10.	Limit aircraft option (if non-zero, pointer to LIMAC)	XX	15	10°	4
		Total bits used (11 excess)			49

Packed Computer Word Diagram

Element	Excess	1	2	3	4	5	6	7	8	9	10
Bits	11	7	7	7	4	6	6	6	1	1	4

IOPTN - Continued

Attribute table IPOPTN (dimensioned 4 by 10) describes the packed word IOPTN.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4	5	6	7	8	9	10
Length	7	7	7	4	6	6	6	1	1	4
Sign	0	0	0	0	0	0	0	0	0	0
Scale	0	0	0	0	0	0	0	0	0	0
First Shift	42	35	28	24	18	12	6	5	4	0

Aircraft Performance Rates. IPERFR

Each aircraft performance class has associated with it a rate of performance for various operational categories.

Operational categories used in the Terminal Area Air Traffic Control Model include:

1. (IPERFR1) - maximum descent rate (hundreds of ft/min)
2. (IPERFR2) - enroute descent rate (hundreds of ft/min)
3. (IPERFR3) - terminal descent rate (hundreds of ft/min)
4. (IPERFR4) - climb rate to 10,000 ft (hundreds of ft/min)
5. (IPERFR5) - climb rate to 20,000 ft (hundreds of ft/min)
6. (IPERFR6) - acceleration rate (kts/min)
7. (IPERFR7) - deceleration rate (unsigned) (kts/min).

Dimension: 3 by 7

Packed words for each entry: 1/8

Aliases: IPERFR(i,1) or IPERFR1(i)
IPERFR(i,2) or IPERFR2(i)
IPERFR(i,3) or IPERFR3(i)
IPERFR(i,4) or IPERFR4(i)
IPERFR(i,5) or IPERFR5(i)
IPERFR(i,6) or IPERFR6(i)
IPERFR(i,7) or IPERFR7(i)

IPERFR

(packed 8 elements in each computer word)

Characteristics of Each Element

Description: Climb, descent, acceleration, or deceleration rates by aircraft performance class (see list above)

Form: XXX

Max. Value: 127

Scale Factor: 10^2 (except IPERFR6 and IPERFR7 are 10°)

Binary Bits: 7

Packed Computer Word Diagram

Element	Excess	1	2	3	4	5	6	7	8
Bits	4	7	7	7	7	7	7	7	7

Uniform Pack/Unpack routines used; no attribute tables required.

Performance Speeds (Kts). IPERFS

Each aircraft performance class has associated with it a speed for various operational categories.

Operational categories used in the Terminal Area Air Traffic Control Model include:

1. (IPERFS1) - Takeoff
2. (IPERFS2) - Climb to 10,000 ft
3. (IPERFS3) - Climb to 20,000 ft
4. (IPERFS4) - Cruising
5. (IPERFS5) - Transition
6. (IPERFS6) - Terminal
7. (IPERFS7) - Approach
8. (IPERFS8) - Final.

Dimension: 4 by 8

Packed words for each entry: 1/6

Aliases: IPERFS(i,1) or IPERFS1(i)
IPERFS(i,2) or IPERFS2(i)
IPERFS(i,3) or IPERFS3(i)
IPERFS(i,4) or IPERFS4(i)
IPERFS(i,5) or IPERFS5(i)
IPERFS(i,6) or IPERFS6(i)
IPERFS(i,7) or IPERFS7(i)
IPERFS(i,8) or IPERFS8(i)

IPERFS

(packed 6 elements in each computer word)

Characteristics of Each Element

Description: Airspeed associated with a performance class--operational category

Form: XXX.

Max. Value: 1023

Scale Factor: 10^0

Binary Bits: 10

Packed Computer Word Diagram

Element	1	2	3	4	5	6
Bits	10	10	10	10	10	10

Uniform Pack/Unpack routines used; no attribute tables required.

Aircraft Performance, Seconds to Lift-Off. IPERFT

Each aircraft performance class has associated with it a performance parameter "seconds to lift-off."

Dimension: 3

Packed word for each entry: 1/10

Aliases: none

IPERFT

(packed 10 elements in each computer word)

Characteristics of Each Element

Description: seconds from start of take-off roll to lift-off

Form: XX.

Max. Value: 63

Scale Factor: 10^0

Binary Bits: 6

Packed Computer Word Diagram

Element	1	2	3	4	5	6	7	8	9	10
Bits	6	6	6	6	6	6	6	6	6	6

Uniform Pack/Unpack routines used; no attribute tables required.

Aircraft Performance, Turn Rate. IPFTR

Each aircraft performance class has associated with it a performance parameter, "turn rate."

Dimension: 3

Packed words for each entry: 1/10

Aliases: none

IPFTR

(packed 10 elements in each computer word)

Characteristics of Each Element

Description: Turn rate (deg/sec)

Form: X.X

Max. Value: 63

Scale Factor: 10^{-1}

Binary Bits: 6

Packed Computer Word Diagram

Element	1	2	3	4	5	6	7	8	9	10
Bits	6	6	6	6	6	6	6	6	6	6

Uniform Pack/Unpack routines used; no attribute tables required.

Length (distance) in n mi of path segments. ISDST

Each flight path segment has an associated distance.

Dimension: 17

Packed words for each entry: 1/6

Aliases: none

ISDST
(packed 6 elements in each computer word)

Characteristics of each element:

Description: distance associated with a segment

Form: XX.X

Max. Value: 1023

Scale Factor: 10^{-1}

Binary Bits: 10

Packed Computer Word Diagram

Element	1	2	3	4	5	6
Bits	10	10	10	10	10	10

Uniform pack/unpack routines used; no attribute tables required.

Schedule of all path segments used for the various paths making up the scheduled routes. ISKED.

Describes the four-dimensional flight plan using both aircraft and ground-based navigational techniques.

Dimension: 85

Packed Words for each entry: 1

Aliases: none

ISKED

Element Number	Description†	Form	Max. Value	Scale Factor	Binary Bits
1.	Flight navigational mode	XX	31	10^0	5
2.	Information No. 1	XXX.	511	10^0	9
3.	Information No. 2	XXX.	±511	10^0	10
4.	Subscript for altitude array (ALTSKD). If value is zero, continue at present altitude; if value is 31, use altitude subscript from IOPTN altitude pointer	XX	31	10^0	5
5.	Subscript to speed type (IPERFS)	XX	15	10^0	4
6.	Control point mode, i.e., type of make good conditions	XX	31	10^0	5
7.	Information No. 3	XXX.	511	10^0	9
8.	Information No. 4	XXXX.	±4095	10^0	13
		Total Bits Used (no excess)			60

Packed Computer Word Diagram

Element	1	2	3	4	5	6	7	8
Bits	5	9	10	5	4	5	9	13

Attribute table IPSKED (dimensioned 4 by 8) describes the packed word ISKED.

†See table on following page for detail.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4	5	6	7	8
Length	5	9	10	5	4	5	9	13
Sign	0	0	1	0	0	0	0	1
Scale	0	0	0	0	0	0	0	0
First Shift	55	46	36	31	27	22	13	0

FLIGHT NAVIGATIONAL MODE (element 1)	INFO. 1 (element 2)	INFO. 2 (element 3)
1 (VOR)	VOR ID	radial (deg) $\begin{cases}\text{+toward}\\ \text{-away}\end{cases}$
2 (ADF)	ADF ID	
3 (VECTOR)	heading (deg)	
4 (STD TURN)	final heading (deg)	direction $\begin{cases}\text{1 left}\\ \text{2 right}\end{cases}$ (rate supplied by program)
5 (DME)	DME ID	distance (n mi) $\begin{cases}\text{+toward}\\ \text{-away}\end{cases}$ (further scale by 10^1 before packing and 10^{-1} after unpacking)
6 (ILS)	ILS ID	radial (deg)
7 (no longer used)		
8 (no longer used)		
9 (TRUE TURN)	direction $\begin{cases}\text{1 left}\\ \text{2 right}\end{cases}$	
10 (VECTOR ON BASE LEG)	heading (deg)	

CONTROL POINT MODE (element 6)		INFO. 3 (element 7)	INFO. 4 (element 8)
1	(RADIAL)	VOR ID	degrees
2	(TIME)	1 2	time of day - scans elapsed time - sec
3	(DIST)	distance (n mi) (further scale by 10^1 before packing and 10^{-1} after unpacking)	
4	(ALT)	hundreds of ft	
5	(FIX)	FIX ID	
6	(SPD)	kts	
7	(DME)	DME ID	distance (n mi) $\begin{cases} +\text{toward} \\ -\text{away} \end{cases}$ (further scale by 10^1 before packing and 10^{-1} after unpacking)
8	(CONTINUE UNTIL CLEARANCE RECEIVED)		
9	(PROCEED IMMEDIATELY TO NEW FLIGHT MODE)		
10	(TOUCHDOWN)	RUNWAY ID	distance (n mi) (further scale by 10^1 before packing and 10^{-1} after unpacking)

Traffic Sample Description. ITRAF1, ITRAF2

Describes the offer time values of the variables associated with each aircraft in the traffic sample.

Dimensions: 102 by 2 per aircraft

Packed words for each entry: 2

Aliases: none

ITRAF1

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	Aircraft ID (generally its relative position in this array)	XXX	1023	10^0	10
2.	x coord. (n mi north)	±XX.XX	±8191	10^{-2}	14
3.	y coord. (n mi east)	±XX.XX	±8191	10^{-2}	14
4.	z coord. (ft above mean sea level)	XXXXX.	65435	10^0	16
		Total Bits Used (6 excess)			54

Packed Computer Word Diagram

Element	Excess	1	2	3	4
Bits	6	10	14	14	16

Attribute table IPTRAF1 (dimensioned 4 by 4) describes the packed word ITRAF1.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4
Length	10	14	14	16
Sign	0	1	1	0
Scale	0	2	2	0
First Shift	44	30	16	0

ITRAF2

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	Speed (kts)	XXX.	511	10^0	9
2.	Heading (deg)	XXX.	511	10^0	9
3.	Aircraft performance class	XX	31	10^0	5
4.	Propeller (=1) or Jet (=2)	X	3	10^0	2
5.	Route number	XX	63	10^0	6
6.	T_o (4-sec scans), i.e., offer time	XXXX	16383	10^0	14
7.	Departure (=1) or arrival (=2)	X	3	10^0	2
8.	Hemisphere relative to terminal, north=1, south=2	XX	15	10^0	4
9.	Desired altitude (hundreds of ft) (departures only)	XXX	511	10^2	9
		Total Bits Used (0 excess)			60

Packed Computer Word Diagram

Element	1	2	3	4	5	6	7	8	9
Bits	9	9	5	2	6	14	2	4	9

Attribute table IPTRAF2 (dimensioned 4 by 9) describes the packed word ITRAF2.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4	5	6	7	8	9
Length	9	9	5	2	6	14	2	4	9
Sign	0	0	0	0	0	0	0	0	0
Scale	0	0	0	0	0	0	0	0	-2
First Shift	51	42	37	35	29	15	13	9	0

Air Traffic Flow Control Descriptor. LIMAC

Describes parameters used to change separation standards for traffic flow control.

Dimension:

Packed words for each entry: 1

Aliases: none

LIMAC

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	Indicator whether limit is on an ETA array or a waiting queue	X	7	10^0	3
2.	Subscript pointing to which ETA or waiting queue array	XXX	63	10^0	6
3.	Upper limit of number of aircraft in array before additional separation applied	XX	63	10^0	6
4.	Lower limit of number of aircraft in array before additional separation removed	XX	63	10^0	6
5.	Separation (n mi) to be added	XX.X	511	10^1	9
6.	Pointer to switch indicating whether additional separation was previously applied	XX	63	10^0	6
7.	Pointer to next possible flow control descriptor. If sign is negative, both conditions must be met to apply additional separation. If positive, either condition being met will use the additional separation	$\pm$XX	$\pm$31	10^0	6
		Total Bits Used (18 excess)			42

Packed Computer Word Diagram

Element	Excess	1	2	3	4	5	6	7
Bits	18	3	6	6	6	9	6	6

The attribute table IPLIMAC (dimensioned 4 by 7) describes the packed word LIMAC.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4	5	6	7
Length	3	6	6	6	9	6	6
Sign	0	0	0	0	0	0	1
Scale	0	0	0	0	1	0	0
First Shift	39	33	27	21	12	6	0

Flight Path (Leg) Description. NNPATH(i)

Describes which segments from ISKED are used for each path.

Dimension: 50

Packed words for each entry: 1

Aliases: none

NNPATH

Element Number	Description	Form	Max. Value	Scale Factor	Binary Bits
1.	Number of segments for this path	X	7	10^0	3
2.	Pointer to next optional vector in NNPATH	XXX	127	10^0	7
3.	Path number for this path (not necessarily equal to "i" and not necessarily unique)	XXX	127	10^0	7
4.	Pointer to ISKED for first segment this path	XXX	127	10^0	7
5.	Pointer to ISKED for second segment this path	XXX	127	10^0	7
6.	Pointer to ISKED for third segment this path	XXX	127	10^0	7
7.	Pointer to ISKED for fourth segment this path	XXX	127	10^0	7
		Total Bits Used (15 excess)			45

Packed Computer Word Diagram

Element	Excess	1	2	3	4	5	6	7
Bits	15	3	7	7	7	7	7	7

Attribute table IPNPATH (dimensioned 4 by 7) describes the packed word NNPATH.

Attribute Table for Pack/Unpack Routines

Element	1	2	3	4	5	6	7
Length	3	7	7	7	7	7	7
Sign	0	0	0	0	0	0	0
Scale	0	0	0	0	0	0	0
First Shift	42	35	28	21	14	7	0

Preceding page blank

APPENDIX G

Computer Core Storage Requirements for the Terminal Area Simulation

This appendix describes how core storage is allocated to the various overlays, subroutines and data arrays. The data storage requirements are variable and are a function of the particular data set with which the program is dealing.

PRECEDING PAGE BLANK NOT FILMED

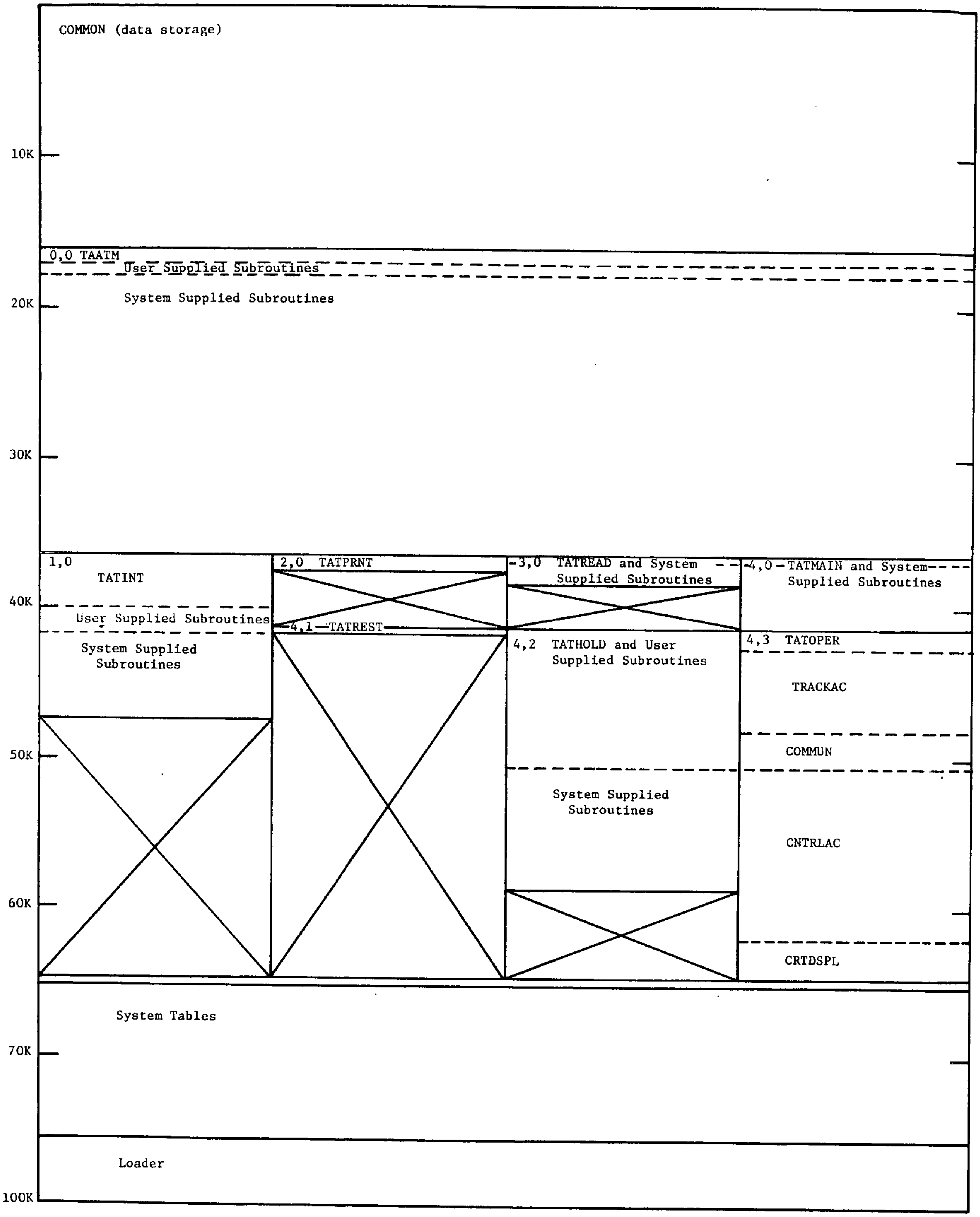
STORAGE REQUIREMENTS

		Octal		Decimal
System reserved storage		100_8		64
Common data storage		14672_8		6586
Overlay 0,0 - resident overlay				
TAATM	512_8		330	
PAKSPL and UPKSPL	266_8		182	
ISHFTE	3_8		3	
PAKUNI	111_8		73	
UPKUNI	76_8		62	
LSHFT	3_8		3	
System Supplied Routines	16751_8		7657	
		20166_8		8310
Overlay 1,0 - shared core				
TATINT	2600_8		1408	
INIT	1344_8		740	
System Supplied Routines	4401_8		2305	
		(10545_8)		(4453)
Overlay 2,0 - shared core				
TATPRNT	556_8		366	
		(556_8)		(366)
Overlay 3,0 - shared cord				
TATREAD	200_8		128	
System Supplied Subroutines	1155_8		621	
		(1355_8)		(749)
Overlay 4,0 - shared core				
TATMAIN	171_8		121	
System Supplied Routines	3440_8		1824	
		3631_8		1945
Overlay 4,1 - shared core				
TATREST	241_8		161	
System supplied routines	100_8		64	
		(341_8)		(225)
Overlay 4,2 - shared core				
TATHOLD	7142_8		3682	
MAKTAPE	100_8		64	
DASHCIR	124_8		84	
System Supplied Routines	6437_8		3359	
		(16025_8)		(7189)

STORAGE REQUIREMENTS, Continued

		Octal		Decimal
Overlay 4,3 - shared core				
TATOPER	1056_8		558	
TRACKAC	4267_8		2231	
RANDU	40_8		32	
COMMUN	1666_8		950	
CNTRLAC	7544_8		3940	
ETACOMP	334_8		220	
CONCHK	363_8		243	
SPDCNTR	373_8		251	
COMETA	116_8		78	
CRTDSPL	1533_8		859	
XFORM	106_8		70	
System supplied routines	256_8		174	
		22606_8		9606
System Tables		10202_8		4226
Loader		3450_8		1832
TOTAL CORE REQUIREMENTS		77471_8		32569

CORE STORAGE



$2K_8$ Per Vertical Division

APPENDIX H

Detailed Logic Flow Charts

This appendix contains the detailed logical flow charts for the simulation model, broken down into overlays and subroutines as follows:

PAGE 1

```
OVERLAY (TAATMFL.0.0)

PROGRAM TAATM(INPUT=201.OUTPUT=201.TAPE10=201.TAPE6=201)

COMMON/OVRBLK/ NTABLE.NINTEG.NLOGIC.NADC.NDAC.NLDISI.NLDISO.
    IJKBUFF(5).    IVARBUF(5).    KTYPE.        VARCHNG

EQUIVALENCE (INTEG(2).IPL).(INTEG(3).ISL)
LOGICAL LDISI. LDISO

COMMON / REALTIM/ ADC(32).DAC(64).LDISI(108).LDISO(196).
NOPER.NHOLD.NRESET.NTERM.NPRINT.NREAD

COMMON/TABLES/
    INTEG(80).     TABLE(075).    LOGIC(20)

LOGICAL LOGIC

*****
* * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN
*****

COMMON/MAJOR1/
    ACLS(50).       ALTOPT(3.7.2). ALTSKD(9).
    ARRMNA.         ARRMXA.        BOUND.
    CGLIDE.         CILS.          CRA20.
    CRA30.          CRB20.         CRB30.
    DGRTD.          DTR.           EHOLDD.
    ENROTA(2).      FTNM.          G.
    GLIDE(2).       HOLDSEP.       IARRN.
    IARRS.          IART(11).      IBF(110).
    ICHKPT(40).     ICNTRA1(85).   ICNTRA2(85).
    IDELT.          IDEPN.         IDEPS.
    IDROUTE(10).    IHPATHE.       IHPATHS.
    INAV(10.5).     IOPTN(55).     IPAPATH(4.5).
    IPASKED(4.7).   IPCHKPT(4.6).  IPCNTR1(4.6).
    IPCNTR2(4.2).   IPORT(4.5).    IPERFR(3.7).
    IPERFS(4.8).    IPERFT(3).     IPETA1(4.4).
    IPETA2(4.6).    IPFTR(3).      IPMSG2(4.6).
    IPMSG3(4.3).    IPNPATH(4.7).  IPNSITE(4.4).
    IPOPL(2).       IPOPPA.        IPOPR(2).
    IPOPTN(4.9).    IPSKED(4.8).   IPTAS(4.5)

COMMON/MAJOR2/
    IPTRAF1(4.4).   IPTRAF2(4.9).  ISDST(17).
    ISKDPU.         ISKED(85).     ISTARTA(21).
    ITANG(2.2).     ITRAF1(102).   ITRAF2(102).
    MAXK.           MF.
    NALTSKD.        NCP.           NDEPR.
    NNAV(5).        NNPATH(50).    NOUT(20).
    NSPDW.          OUT(26).       PI.
    PPMX(3).        PPMY(3).       RCP(2).
    RTD.            RWYX(4.2).     RWYY(4.2).
    RWYZ(2).        SHOLDD.        SPOSTD(9.2).
    SQX(6).         SQY(6).        TRIX(5).
    TRIY(5).        XA(2).         XC(2).
    XDILS(2).       XDIST(2).      XFINAL(2).
    XRPOS(2).       YA(2).         YC(2).
    YRPOS(2).       RAD(3).        SIG(3).        IARG
  . IETCOOR(36).    IPETC(4.2).    IACFPER(3)
```

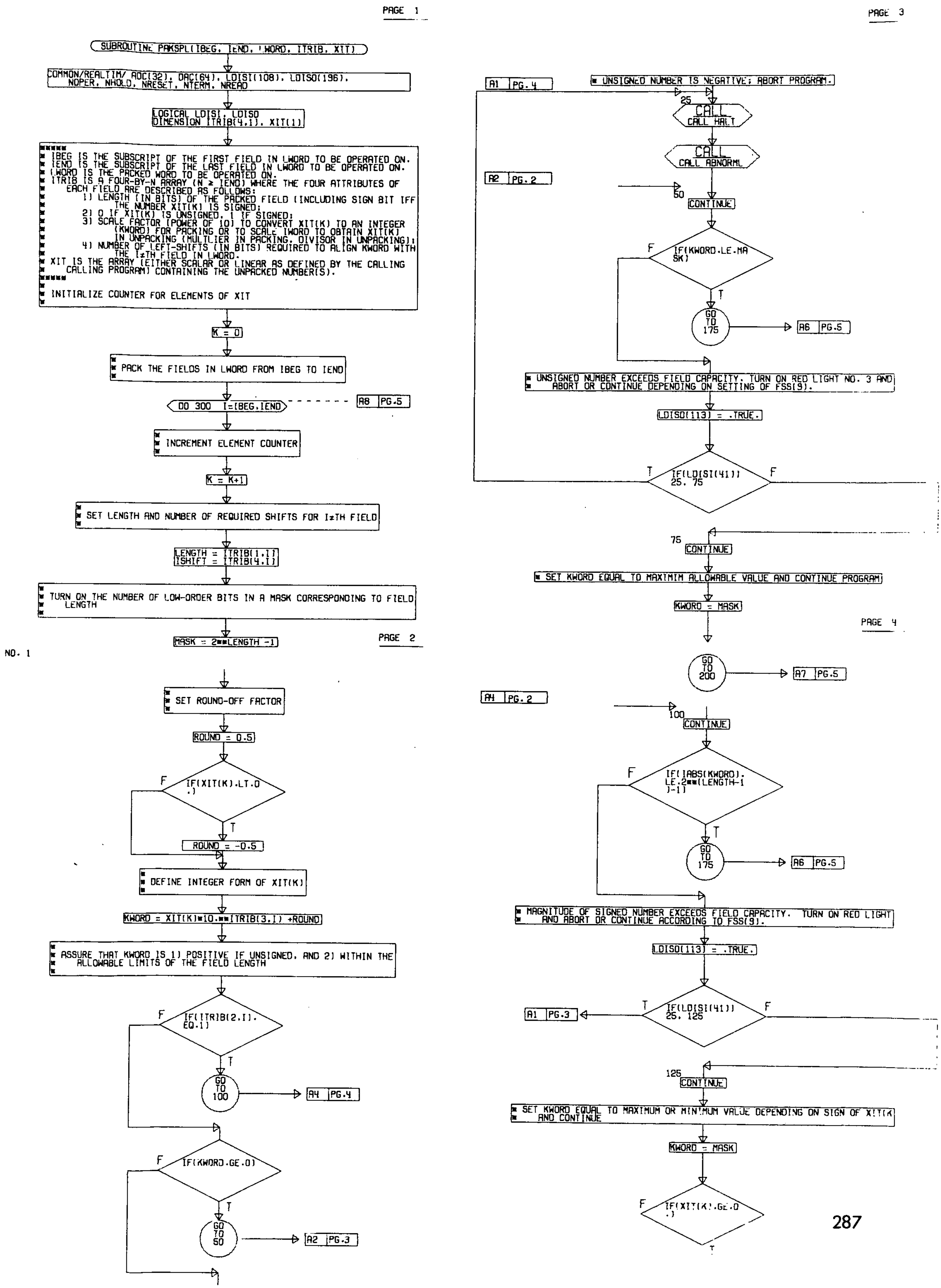
CROSSHAIR NO. 1

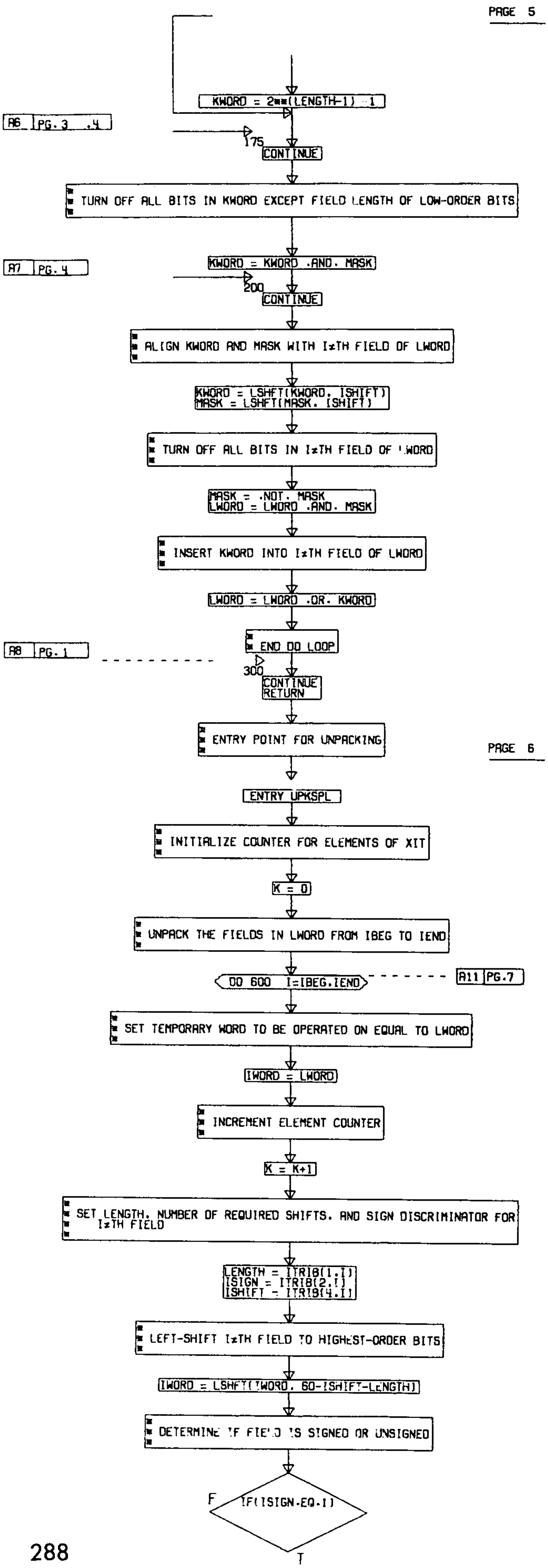
PAGE 2

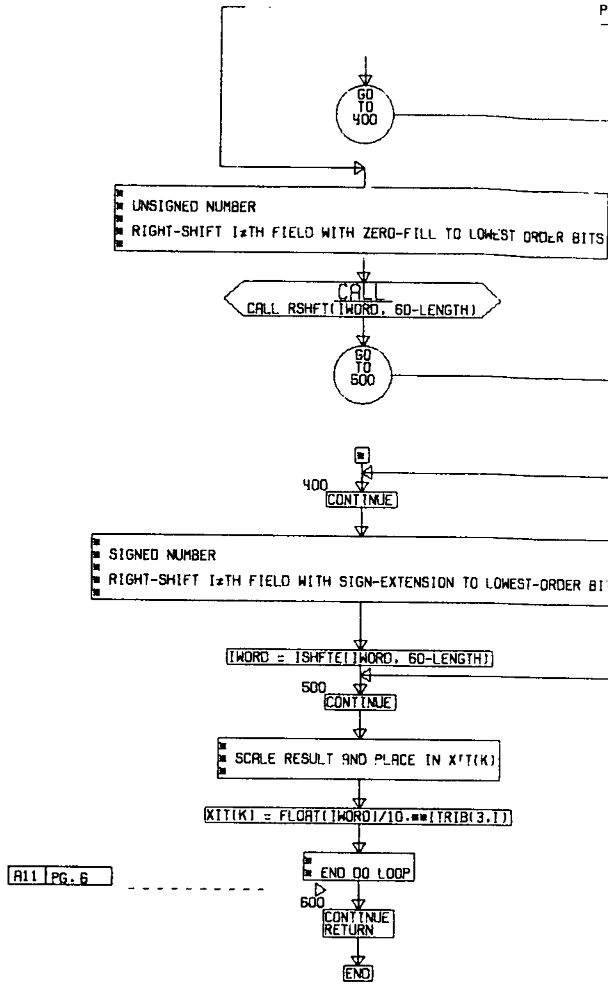
```
*****
*****
* * * MISCELLANEOUS PRIMARY-TO-PRIMARY AND PRIMARY-TO-SECONDARY COMMON
*****       (0.0). (1.0). (2.0). (3.0). (4.0). (4.1). (4.2). (4.3)

COMMON/MISC/
    DPRIM.          IIHEM(21).     ITL.          NFRAMF.
    PITCH

*****

COMMON/TOES/ IONE.ITWO.ITHREE.IFOUR.IFIVE.ISIX.ISEVEN.IEIGHT.ININE
.ITEN

*****
* * * REAL-TIME MAIN COMMON IN OVERLAY(4.0) AND ALL SECONDARY OVERLAYS A
* * *     PRIMARY SUBROUTINES OF (4.0)
*****

COMMON/RTMAIN1/
   ASKSPD(35.11).ASPACE1(4.9).ASPACE2(4.9).ASPACE3(4.9).
    CETA.           CONFLCT.       DELH(35).
    HDBEG(35).      HDOWN.         IA(2).
    IAC.            IACTN1(35).    IACTN2(35).
 IACTN3(35). IASKED(35.11). IATAS(35.2).ICNGO.
    IATRAF(35.26).  ICA.           ICACT.
    ICLASS.         ICMSG1(8.5).   ICMSG2(8.5).
    ICMSG3(8.5).    ICOMM(10).     ID(2).
    IDELAY.         IENROT(2.20).  IETAS(35.2).
    IETA1(36.10).   IETA2(36.10).  IE360(60).
    IFCMSG1(8.25).  IFCMSG2(8.25). IFCMSG3(8.25).
    IHOWN.          IOBETA.        IOEETA.
    IOPATH.         IPASSAC.       ISEQ.
    ISEQUE(3.16).   ISW4.          ITOQUE(2.20).
    ITRY.           IWHICH.        KNTP.
    L.              LEMG(5).       NASPACE(4).
    NCMSG(10).      NEMG.          NETA(36).
    NFMSG(10).      NHOLDP(35).    NNA(35).
    NSEQ.           OLDR(35)

COMMON/RTMAIN2/
    OTHAA.          OTHETA.        OTHOA.
    OTHSETA.        RESTOR(35).    ROLL.
    T(35).          TETAE.         TETAL.
    TETAO.          TIME.          V1.
    V2.             XOWN.          XPL(35).
    YOWN.           YPL(35).       ZOWN
  . R9LX(2).        R9LY(2).       R9RX(2).       R9RY(2).
    R15X(2).        R15Y(2).       R21X(2).       R21Y(2).
    XNREF(4).       YNREF(4)

*****
*     SPECIAL COMMON FOR CNTRLAC
*****
```

PAGE

```
COMMON/RTMAIN3/
    ALTLIM.         DALT.          DELAYT.
    HOLDTIM.        HT1.           HT2.
    IACTPT.         ICNT.          ICONTRL.
    IDENT.          IDVERG.        IETT.          IDEST.
    IHEM.           ILEG.          INEXT.
    IRCLR.          IS.            ITEMP.
    ITEMPA(10).     ITEMPS.        ITRYDVG.
    ITRYEH.         ITRYSH.        ITRYSPD.
    ITRYVEC.        ITYPCL.        ITYPE.
    IVARSPD.        IVECT2.        IWCHAQ.
    IWHCH.          IWHCNTR.       IWHSPD.
    J.              JGO.           JJ.
    JJJ.            JL.            JTEMPA(10).
    K.              KGO.           KALTOP.
    KK.             K3.            LIND.
    NCLRD.          NOM.           SAVE(35).
    SAVE2(35).      SPDSAV.        TETAOM.
    TMSGA(6).       TMSGA2(3).     RCFNM(2)

COMMON/RTMAIN4/
 A. ANGLE.    ANG2.     B.        I.        IAALT.    IANG.
    IAT.      ICIP1.    ICONTR.   ICP.      IDT.      IETT2.
    IGO.      IH.       ILEGM1.   IND.      INOR.     IPA.
    IPD.      IRSEP.    IRT.      ISA.      ISD.      ISEQM1.
    ISUBSPD.  ITEMP2.   IWAC.     IWETA.    IWHP1.    IWID.
    JJM1.     JJP1.     JOE.      JP1.      JS.       JTEMP.
    ::.       LL'.      LLLP3.    M.        MM.       MMM.
    MTEMP.    NCMSGT.   NFMSGT.   NTEMP.    N360.     RATE.
    RC.       RDESC.    R1.       SEPCOR.   SPEED.    TAFIX.
    TEMETA.   TEMGAT.   TEMOFF.   TEMONN.   TEMP.     TEMPA3(10).
    TEMSEP.   TTH.      TXA.      TXC.      TXI.      TYA.
    TYC.      TYI.      XB.       XCLR2.    XI.       XLID.
    XLTT2.    XMSGL2.   YB.       YI

*****

CALL
CALL RTROUTE(MF.90034S)

OLFN=7LTAATMFL

CALL
CALL PCCEXEC(OLFN.MF.90034.LDISI.IPL.ISL)

CALL
CALL ROLFN(IJKBUFF(1))

END
```


PAGE 1
SUBROUTINE PAKSPL(IBEG, IEND, LWORD, ITRIB, XIT)
COMMON/REALTIM/ ADC(32), DAC(64), LDISI(108), LDISO(196), NOPER, NHOLD, NRESET, NTERM, NREAD
LOGICAL LDISI, LDISO
DIMENSION ITRIB(4,1), XIT(1)
IBEG IS THE SUBSCRIPT OF THE FIRST FIELD IN LWORD TO BE OPERATED ON.
IEND IS THE SUBSCRIPT OF THE LAST FIELD IN LWORD TO BE OPERATED ON.
LWORD IS THE PACKED WORD TO BE OPERATED ON.
ITRIB IS A FOUR-BY-N ARRAY (N ≥ IEND) WHERE THE FOUR ATTRIBUTES OF EACH FIELD ARE DESCRIBED AS FOLLOWS:
1) LENGTH (IN BITS) OF THE PACKED FIELD (INCLUDING SIGN BIT IFF THE NUMBER XIT(K) IS SIGNED;
2) 0 IF XIT(K) IS UNSIGNED, 1 IF SIGNED;
3) SCALE FACTOR (POWER OF 10) TO CONVERT XIT(K) TO AN INTEGER (KWORD) FOR PACKING OR TO SCALE IWORD TO OBTAIN XIT(K) IN UNPACKING (MULTLIER IN PACKING, DIVISOR IN UNPACKING);
4) NUMBER OF LEFT-SHIFTS (IN BITS) REQUIRED TO ALIGN KWORD WITH THE I≠TH FIELD IN LWORD.
XIT IS THE ARRAY (EITHER SCALAR OR LINEAR AS DEFINED BY THE CALLING CALLING PROGRAM) CONTAINING THE UNPACKED NUMBER(S).
INITIALIZE COUNTER FOR ELEMENTS OF XIT
K = 0
PACK THE FIELDS IN LWORD FROM IBEG TO IEND
DO 300 I=IBEG,IEND
A8 PG.5
INCREMENT ELEMENT COUNTER
K = K+1
SET LENGTH AND NUMBER OF REQUIRED SHIFTS FOR I≠TH FIELD
LENGTH = ITRIB(1,I)
ISHIFT = ITRIB(4,I)
TURN ON THE NUMBER OF LOW-ORDER BITS IN A MASK CORRESPONDING TO FIELD LENGTH
MASK = 2**LENGTH -1
PAGE 2
NO. 1
SET ROUND-OFF FACTOR
ROUND = 0.5
F
IF(XIT(K).LT.0.)
T
ROUND = -0.5
DEFINE INTEGER FORM OF XIT(K)
KWORD = XIT(K)*10.**ITRIB(3,I) +ROUND
ASSURE THAT KWORD IS 1) POSITIVE IF UNSIGNED, AND 2) WITHIN THE ALLOWABLE LIMITS OF THE FIELD LENGTH
F
IF(ITRIB(2,I).EQ.1)
T
GO TO 100
A4 PG.4
F
IF(KWORD.GE.0)
T
GO TO 50
A2 PG.3
PAGE 3
A1 PG.4
UNSIGNED NUMBER IS NEGATIVE; ABORT PROGRAM.
25
CALL
CALL HALT
CALL
CALL ABNORML
A2 PG.2
50
CONTINUE
F
IF(KWORD.LE.MASK)
T
GO TO 175
A6 PG.5
UNSIGNED NUMBER EXCEEDS FIELD CAPACITY. TURN ON RED LIGHT NO. 3 AND ABORT OR CONTINUE DEPENDING ON SETTING OF FSS(9).
LDISO(113) = .TRUE.
T
IF(LDISI(41)) 25, 75
F
75
CONTINUE
SET KWORD EQUAL TO MAXIMIM ALLOWABLE VALUE AND CONTINUE PROGRAM
KWORD = MASK
PAGE 4
GO TO 200
A7 PG.5
A4 PG.2
100
CONTINUE
F
IF(IABS(KWORD).LE.2**(LENGTH-1)-1)
T
GO TO 175
A6 PG.5
MAGNITUDE OF SIGNED NUMBER EXCEEDS FIELD CAPACITY. TURN ON RED LIGHT AND ABORT OR CONTINUE ACCORDING TO FSS(9).
LDISO(113) = .TRUE.
A1 PG.3
T
IF(LDISI(41)) 25, 125
F
125
CONTINUE
SET KWORD EQUAL TO MAXIMUM OR MINIMUM VALUE DEPENDING ON SIGN OF XIT(K) AND CONTINUE
KWORD = MASK
F
IF(XIT(K).GE.0.)
T

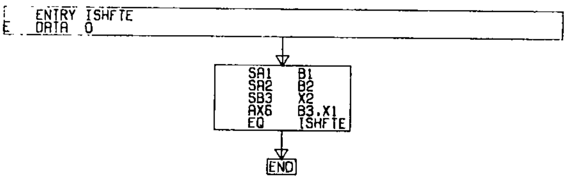

PAGE 5
KWORD = 2**(LENGTH-1) -1
A6 PG. 3 ,4
175
CONTINUE
TURN OFF ALL BITS IN KWORD EXCEPT FIELD LENGTH OF LOW-ORDER BITS
A7 PG. 4
KWORD = KWORD .AND. MASK
200
CONTINUE
ALIGN KWORD AND MASK WITH I*TH FIELD OF LWORD
KWORD = LSHFT(KWORD, ISHIFT)
MASK = LSHFT(MASK, ISHIFT)
TURN OFF ALL BITS IN I*TH FIELD OF LWORD
MASK = .NOT. MASK
LWORD = LWORD .AND. MASK
INSERT KWORD INTO I*TH FIELD OF LWORD
LWORD = LWORD .OR. KWORD
A8 PG. 1
END DO LOOP
300
CONTINUE
RETURN
ENTRY POINT FOR UNPACKING
PAGE 6
ENTRY UPKSPL
INITIALIZE COUNTER FOR ELEMENTS OF XIT
K = 0
UNPACK THE FIELDS IN LWORD FROM IBEG TO IEND
DO 600 I=IBEG,IEND
A11 PG.7
SET TEMPORARY WORD TO BE OPERATED ON EQUAL TO LWORD
IWORD = LWORD
INCREMENT ELEMENT COUNTER
K = K+1
SET LENGTH, NUMBER OF REQUIRED SHIFTS, AND SIGN DISCRIMINATOR FOR I*TH FIELD
LENGTH = ITRIB(1,I)
ISIGN = ITRIB(2,I)
ISHIFT = ITRIB(4,I)
LEFT-SHIFT I*TH FIELD TO HIGHEST-ORDER BITS
IWORD = LSHFT(IWORD, 60-ISHIFT-LENGTH)
DETERMINE IF FIELD IS SIGNED OR UNSIGNED
F
IF(ISIGN.EQ.1)
T

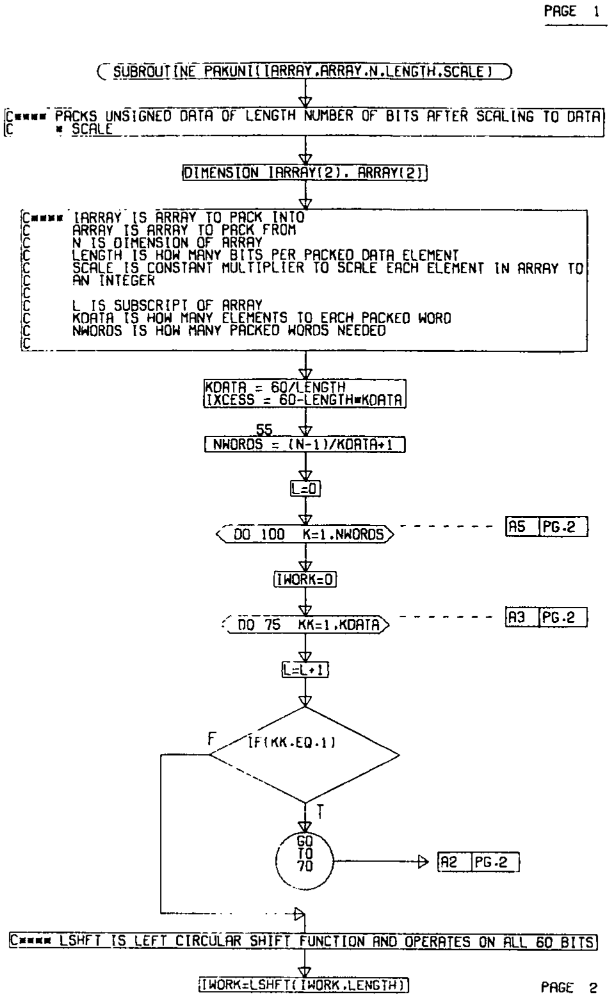

GO TO 400
UNSIGNED NUMBER
RIGHT-SHIFT I*TH FIELD WITH ZERO-FILL TO LOWEST ORDER BITS
CALL
CALL RSHFT(IWORD, 60-LENGTH)
GO TO 600
400
CONTINUE
SIGNED NUMBER
RIGHT-SHIFT I*TH FIELD WITH SIGN-EXTENSION TO LOWEST-ORDER BI
IWORD = ISHFTE(IWORD, 60-LENGTH)
500
CONTINUE
SCALE RESULT AND PLACE IN XIT(K)
XIT(K) = FLOAT(IWORD)/10.**ITRIB(3,I)
A11 PG. 6
END DO LOOP
600
CONTINUE
RETURN
END

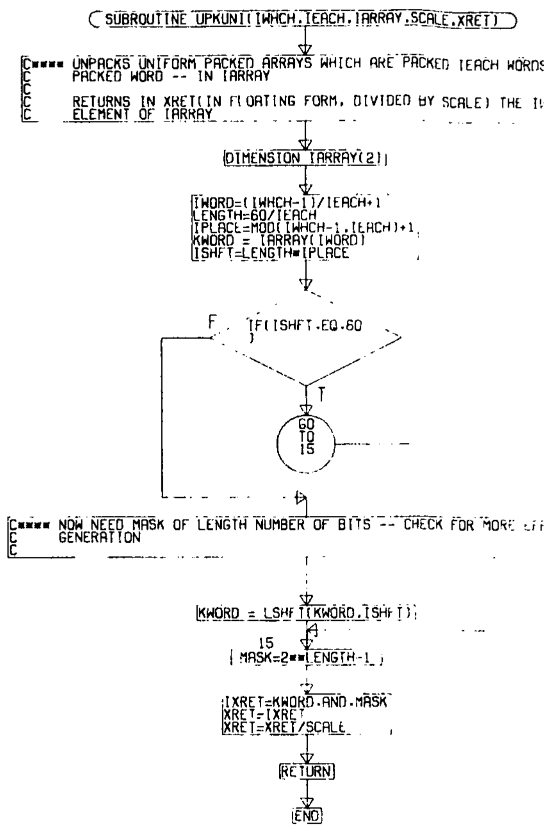
PAGE 1

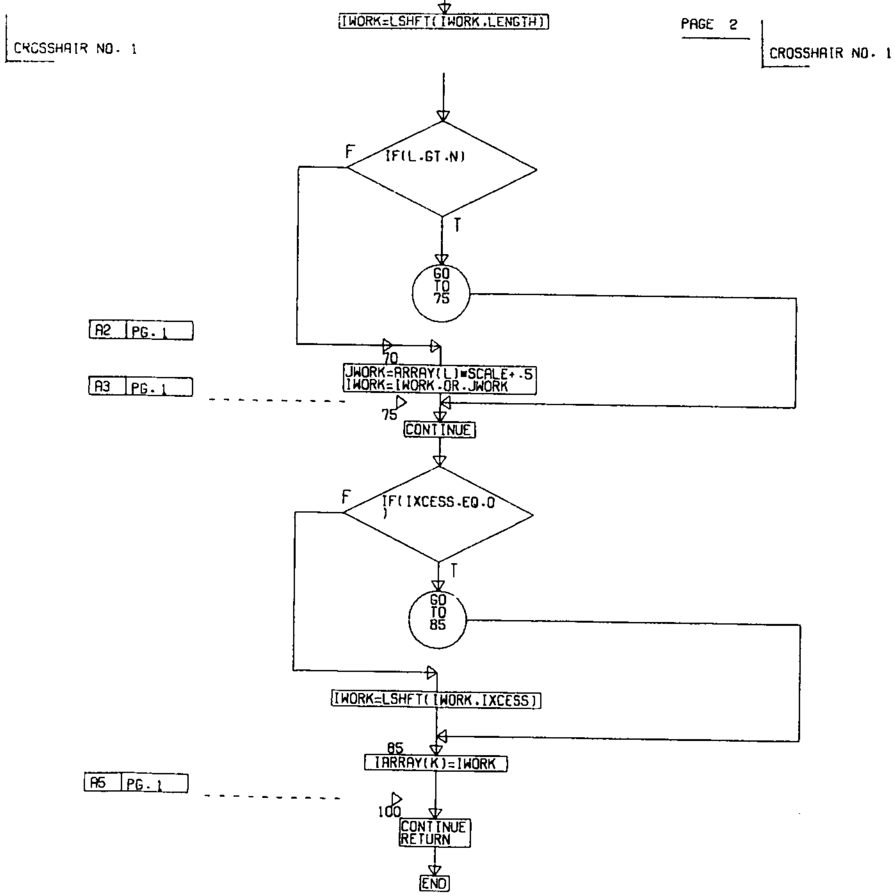
PAGE 2

NO. 1




PAGE 1
SUBROUTINE PAKUNI(IARRAY,ARRAY,N,LENGTH,SCALE)
C***** PACKS UNSIGNED DATA OF LENGTH NUMBER OF BITS AFTER SCALING TO DATA
C * SCALE
DIMENSION IARRAY(2), ARRAY(2)
C***** IARRAY IS ARRAY TO PACK INTO
C ARRAY IS ARRAY TO PACK FROM
C N IS DIMENSION OF ARRAY
C LENGTH IS HOW MANY BITS PER PACKED DATA ELEMENT
C SCALE IS CONSTANT MULTIPLIER TO SCALE EACH ELEMENT IN ARRAY TO
C AN INTEGER
C L IS SUBSCRIPT OF ARRAY
C KDATA IS HOW MANY ELEMENTS TO EACH PACKED WORD
C NWORDS IS HOW MANY PACKED WORDS NEEDED
KDATA = 60/LENGTH
IXCESS = 60-LENGTH*KDATA
55
NWORDS = (N-1)/KDATA+1
L=0
DO 100 K=1,NWORDS
A5 PG.2
IWORK=0
DO 75 KK=1,KDATA
A3 PG.2
L=L+1
F
IF(KK.EQ.1)
T
GO TO 70
A2 PG.2
C***** LSHFT IS LEFT CIRCULAR SHIFT FUNCTION AND OPERATES ON ALL 60 BITS
IWORK=LSHFT(IWORK,LENGTH)


SUBROUTINE UPKUNI(IWHCH,IEACH,IARRAY,SCALE,XRET)
C***** UNPACKS UNIFORM PACKED ARRAYS WHICH ARE PACKED IEACH WORDS
C PACKED WORD -- IN IARRAY
C RETURNS IN XRET(IN FLOATING FORM, DIVIDED BY SCALE) THE I
C ELEMENT OF IARRAY
DIMENSION IARRAY(2)
IWORD=(IWHCH-1)/IEACH+1
LENGTH=60/IEACH
IPLACE=MOD(IWHCH-1,IEACH)+1
KWORD = IARRAY(IWORD)
ISHFT=LENGTH*IPLACE
F
IF(ISHFT.EQ.60
)
T
GO TO 15
C***** NOW NEED MASK OF LENGTH NUMBER OF BITS -- CHECK FOR MORE
C GENERATION
KWORD = LSHFT(KWORD,ISHFT)
15
MASK=2**LENGTH-1
IXRET=KWORD.AND.MASK
XRET=IXRET
XRET=XRET/SCALE
RETURN
END


CROSSHAIR NO. 1
PAGE 2
CROSSHAIR NO. 1
F
IF(L.GT.N)
T
GO TO 75
A2 PG.1
70
JWORK=ARRAY(L)*SCALE+.5
IWORK=IWORK.OR.JWORK
A3 PG.1
75
CONTINUE
F
IF(IXCESS.EQ.0
)
T
GO TO 85
IWORK=LSHFT(IWORK,IXCESS)
85
IARRAY(K)=IWORK
A5 PG.1
100
CONTINUE
RETURN
END

```
IDENT LSHFT
```


NO. 1

```
ENTRY LSHFT
DATA  0

SA1   B1
SA2   B2
SB3   X2
LX6   B3.X1
EQ    LSHFT

END
```

```
OVERLAY (TAATMFL,1,0)
PROGRAM TATINT

COMMON/OVRBLK/ NTABLE,NINTEG,NLOGIC,NADC,NDAC,NLDISI,NLDISO,
   IJKBUFF(5),    IVARBUF(5),    KTYPE,          VARCHNG

EQUIVALENCE (INTEG(2),IPL),(INTEG(3),ISL)
LOGICAL LDISI, LDISO

COMMON / REALTIM/ ADC(32),DAC(64),LDISI(108),LDISO(196),
NAPER,NHOLD,NRESET,NTERM,NPRINT,NREAD

COMMON/TABLES/
   INTEG(80),     TABLE(075),    LOGIC(20)

LOGICAL LOGIC

*****
* * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN
*****

COMMON/MAJOR1/
     ACLS(50),       ALTOPT(3,7,2),  ALTSKD(9),
     ARRMNA,         ARRMXA,         BOUND,
     CGLIDE,         CILS,           CRA20,
     CRA30,          CRB20,          CRB30,
     DGRTD,          DTR,            EHOLDD,
     ENROTA(2),      FINM,           G,
     GLIDE(2),       HOLDSEP,        IARRN,
     IARRS,          IART(11),       IBF(110),
     ICHKPT(40),     ICNTRA1(85),    ICNTRA2(85),
     IDELT,          IDEPN,          IDEPS,
     IDROUTE(10),    IHPATHE,        IHPATHS,
     INAV(10,5),     IOPTN(55),      IPAPATH(4,5),
     IPASKED(4,7),   IPCHKPT(4,6),   IPCNTR1(4,6),
     IPCNTR2(4,2),   IPDRT(4,5),     IPERFR(3,7),
     IPERFS(4,8),    IPERFT(3),      IPETA1(4,4),
     IPETA2(4,6),    IPFTR(3),       IPMSG2(4,6),
     IPMSG3(4,3),    IPNPATH(4,7),   IPNSITE(4,4),
     IPOPL(2),       IPOPPA,         IPOPR(2),
     IPOPTN(4,9),    IPSKED(4,8),    IPTAS(4,5)

COMMON/MAJOR2/
     IPTRAF1(4,4),   IPTRAF2(4,9),   ISDST(17),
     ISKDPU,         ISKED(85),      ISTARTA(21),
     ITANG(2,2),     ITRAF1(102),    ITRAF2(102),
     MAXK,           MF,
     NALTSKD,        NCP,            NDEPR,
     NNAV(5),        NNPATH(50),     NOUT(20),
     NSPDW,          OUT(26),        PI,
     PPMX(3),        PPMY(3),        RCP(2),
     RTD,            RWYX(4,2),      RWYY(4,2),
     RWYZ(2),        SHOLDD,         SPOSTD(9,2),
     SQX(6),         SQY(6),         TRIX(5),
     TRIY(5),        XA(2),          XC(2),
     XDILS(2),       XDIST(2),       XFINAL(2),
     XRPOS(2),       YA(2),          YC(2),
     YRPOS(2),       RAD(3),         SIG(3),      IARG
   , IETCOOR(36),    IPETC(4,2),     IACFPER(3)
```


CROSSHAIR NO. 1

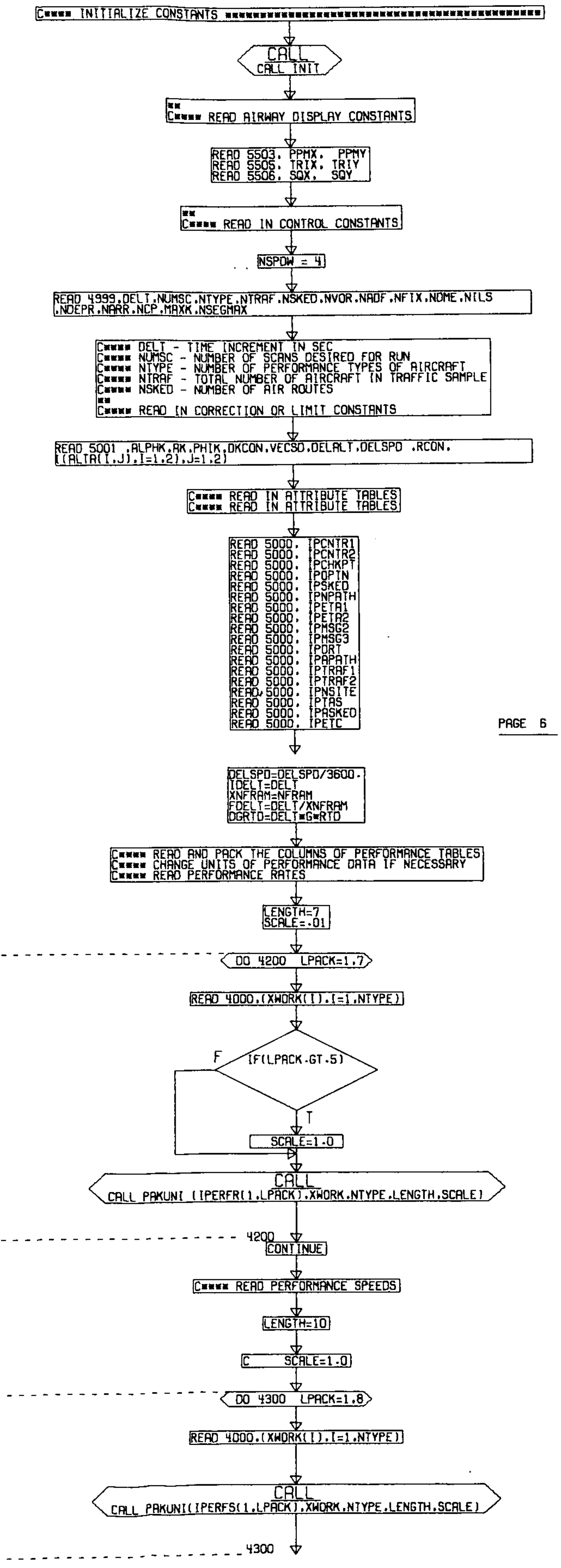
```
*****
*****
* * * MISCELLANEOUS PRIMARY-TO-PRIMARY AND PRIMARY-TO-SECONDARY COMMON
*****     (0,0), (1,0), (2,0), (3,0), (4,0), (4,1), (4,2), (4,3)

COMMON/MISC/
     OPRIM,          IIHEM(21),      ITL,        NFRAMF,
     PITCH

*****

COMMON/TOES/ IONE,ITWO,ITHREE,IFOUR,IFIVE,ISIX,ISEVEN,IEIGHT,ININE
,ITEN

*****
* * * DIMENSIONS AND TYPES FOR OVERLAY(1,0)
*****

DIMENSION
     XL1(2),         XWORK(150),     YWORK(10),
     ZWORK(14,3)

LOGICAL
     VARCHNG

*****
* * * DIM. + TYPE  FOR OVERLAY(1,0)

DIMENSION
     MS6(20)   , ALTA(2,2) , SEPP(2)

* * * TABLE INTEG LOGIC EQIV. FOR OVERLAY(1,0)

EQUIVALENCE
     (INTEG(12),     IOUT)     ,(INTEG(19),IRUN),(INTEG(37),NTRAF)
    ,(INTEG(36),     NFRAM)    ,(INTEG(40),IT),(INTEG(39),NUMSC)
    ,(INTEG(41),     NSCAN)    ,(INTEG(46),     MSG(1))
    ,(TABLE(01),     AK)       ,(TABLE(02),     ALPHK)
    ,(TABLE(03),ALTA(1,1))     ,(TABLE(11),     TEMPS)
    ,(TABLE(12),     VECERR)   ,(TABLE(14),     DEZALT)
    ,(TABLE(15),     DELSPD)   ,(TABLE(16),     DELT)
    ,(TABLE(19),     DKCON)    ,(TABLE(22),     HOLDT)
    ,(TABLE(23),     PASEP)    ,(TABLE(24),     POSEP)
    ,(TABLE(25),     PHIK)     ,(TABLE(26),     BETA)
    ,(TABLE(37),     RCON)     ,(TABLE(38),     SASEP)
    ,(TABLE(43), SEPP(1))      ,(TABLE(48),     VECSD)
    ,(TABLE(53),     CKROLL)   ,(TABLE(54),     CKRC)
    ,(TABLE(55),     CROLL)    ,(TABLE(56),     CRC)
    ,(TABLE(57),     FDELT)    ,(TABLE(58),     ZCON)
    ,(TABLE(59),     HOCON)

*****
* * * DIMENSIONS AND EQUIVALENCES FOR OVERLAY(1,0)
*****
```



```
DIMENSION
     IPERFS4(4),     FSS(16)

LOGICAL FSS

EQUIVALENCE
     (IPERFS(1,4),   IPERFS4(1)),    (LDISI(33),     FSS(1)),
     (NNAV(1),       NVOR),          (NNAV(2),       NADF),
     (NNAV(3),       NFIX),          (NNAV(4),       NDME),
     (NNAV(5),       NILS)

*****

NTABLE=75
NINTEG=80
NLOGIC=20
NADC=13
NDAC=24
NLDISI=48
NLDISO=180

CALL
CALL DATABLX(TABLE,NTABLE,INTEG,NINTEG,LOGIC,NLOGIC,ADC,NADC,DAC,
     NDAC,LDISI,NLDISI,LDISO,NLDISO)

C**** SET UP REAL TIME RECORDING FILE

CALL
CALL READOUT(1,1,IT)

CALL
CALL READOUT(1,1,NSCAN)

CALL
CALL READOUT(1,1,IRUN)

CALL
CALL READOUT(1,1,IOUT)

CALL
CALL READOUT(-20,1,NOUT)

CALL
CALL READOUT(-26,1,OUT)

C**** SET ALL LOGICAL OUTPUTS TO FALSE

DO 90007  INDEX=1,196                    A1  PG.4

A1  PG.3

90007
LDISO(INDEX) =.FALSE.

DO 90008  I=1,108

90008
LDISI(I) = .FALSE.

* * * SET WHITE LIGHTS TO INDICATE OVERLAY COMBINATIONS IN EXECUTIO

LDISO(62)=LDISO(63)=LDISO(64)=LDISO(81)=LDISO(82)=LDISO(83)=.F.
LDISO(61)=.TRUE.
OLFN=7LTAATMFL
IPL=4
ISL=0

CALL
CALL OVLCOPY(IJKBUFF(1),OLFN,IPL,ISL)

C**** SET UP INPUT AND OUTPUT TO REQUESTED EQUIPMENT

CALL
CALL INOUT(ADC,13,DAC,24,LDISI,48,LDISO,180)

CALL
CALL XDSPLAY(LDISI,LDISO,VARCHNG ,KTYPE,IVARBUF,LDISI(48))

T   IF(SENSE SWITC   F
    H 6) 50900,
    50800

50800
CONTINUE

CALL
CALL NM218 (SLOSCAR)

50900
CONTINUE
```

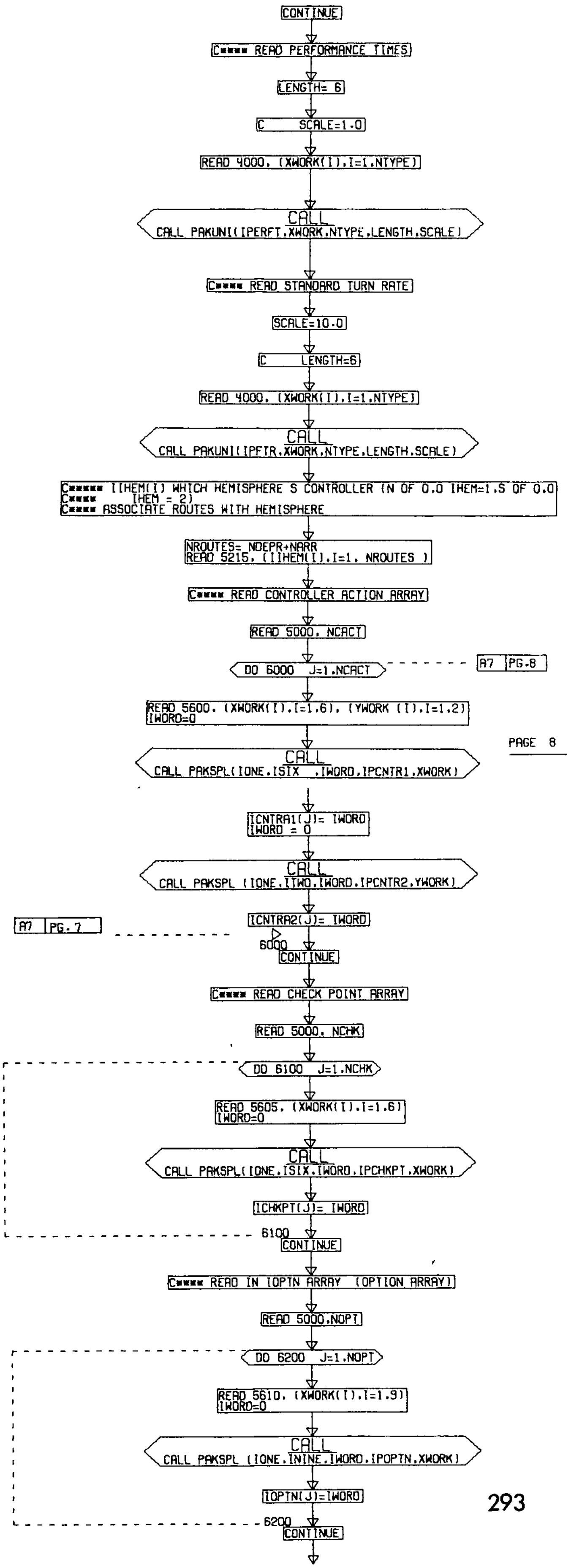

4

PAGE 5

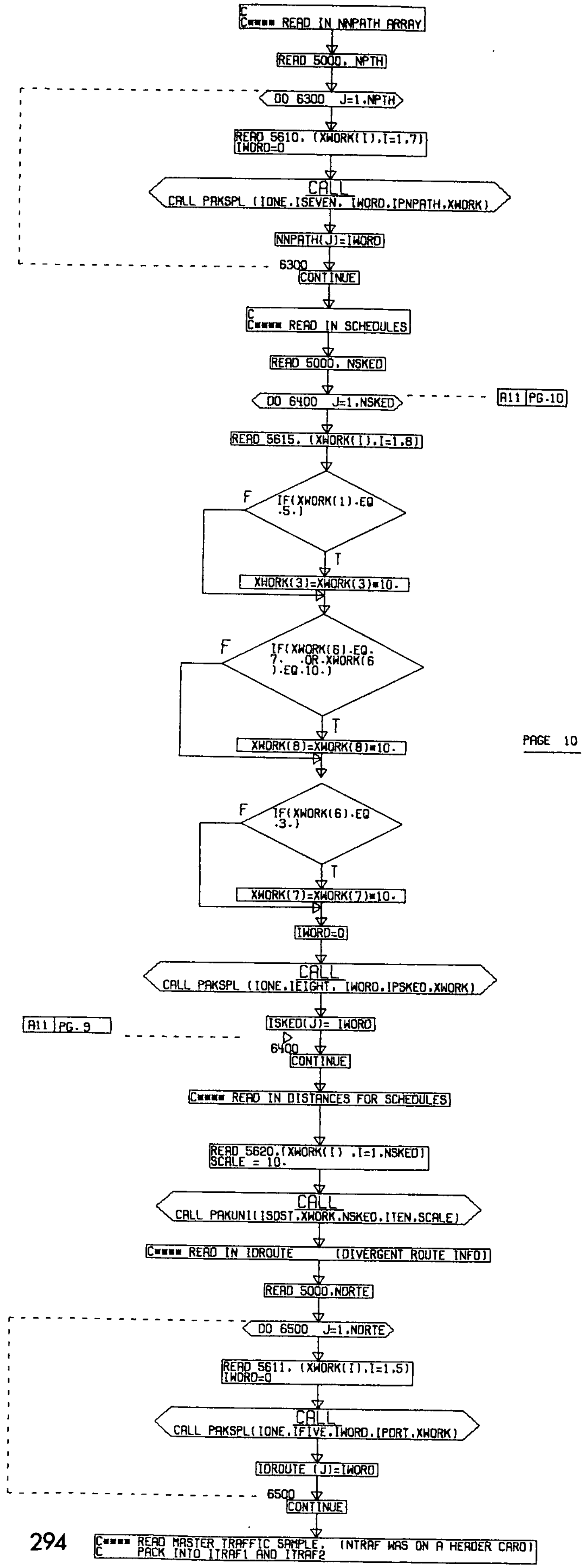


C***** INITIALIZE CONSTANTS **
CALL
CALL INIT
**
C***** READ AIRWAY DISPLAY CONSTANTS
READ 5503, PPMX, PPMY
READ 5505, TRIX, TRIY
READ 5506, SQX, SQY
**
C***** READ IN CONTROL CONSTANTS
NSPDW = 4
READ 4999,DELT,NUMSC,NTYPE,NTRAF,NSKED,NVOR,NADF,NFIX,NDME,NILS
,NDEPR,NARR,NCP,MAXK,NSEGMAX
C***** DELT - TIME INCREMENT IN SEC
C***** NUMSC - NUMBER OF SCANS DESIRED FOR RUN
C***** NTYPE - NUMBER OF PERFORMANCE TYPES OF AIRCRAFT
C***** NTRAF - TOTAL NUMBER OF AIRCRAFT IN TRAFFIC SAMPLE
C***** NSKED - NUMBER OF AIR ROUTES
**
C***** READ IN CORRECTION OR LIMIT CONSTANTS
READ 5001 ,ALPHK,AK,PHIK,DKCON,VECSO,DELALT,DELSPD ,RCON,
((ALTA(I,J),I=1,2),J=1,2)
C***** READ IN ATTRIBUTE TABLES
C***** READ IN ATTRIBUTE TABLES
READ 5000, IPCNTR1
READ 5000, IPCNTR2
READ 5000, IPCHKPT
READ 5000, IPOPTN
READ 5000, IPSKED
READ 5000, IPNPATH
READ 5000, IPETA1
READ 5000, IPETA2
READ 5000, IPMSG2
READ 5000, IPMSG3
READ 5000, IPORT
READ 5000, IPAPATH
READ 5000, IPTRAF1
READ 5000, IPTRAF2
READ 5000, IPNSITE
READ 5000, IPTAS
READ 5000, IPASKED
READ 5000, IPETC
PAGE 6
DELSPD=DELSPD/3600.
IDELT=DELT
XNFRAM=NFRAM
FDELT=DELT/XNFRAM
DGRTD=DELT*G*RTD
C***** READ AND PACK THE COLUMNS OF PERFORMANCE TABLES
C***** CHANGE UNITS OF PERFORMANCE DATA IF NECESSARY
C***** READ PERFORMANCE RATES
LENGTH=7
SCALE=.01
DO 4200 LPACK=1,7
READ 4000,(XWORK(I),I=1,NTYPE)
IF(LPACK.GT.5)
F
T
SCALE=1.0
CALL
CALL PAKUNI (IPERFR(1,LPACK),XWORK,NTYPE,LENGTH,SCALE)
4200
CONTINUE
C***** READ PERFORMANCE SPEEDS
LENGTH=10
C SCALE=1.0
DO 4300 LPACK=1,8
READ 4000,(XWORK(I),I=1,NTYPE)
CALL
CALL PAKUNI(IPERFS(1,LPACK),XWORK,NTYPE,LENGTH,SCALE)
4300

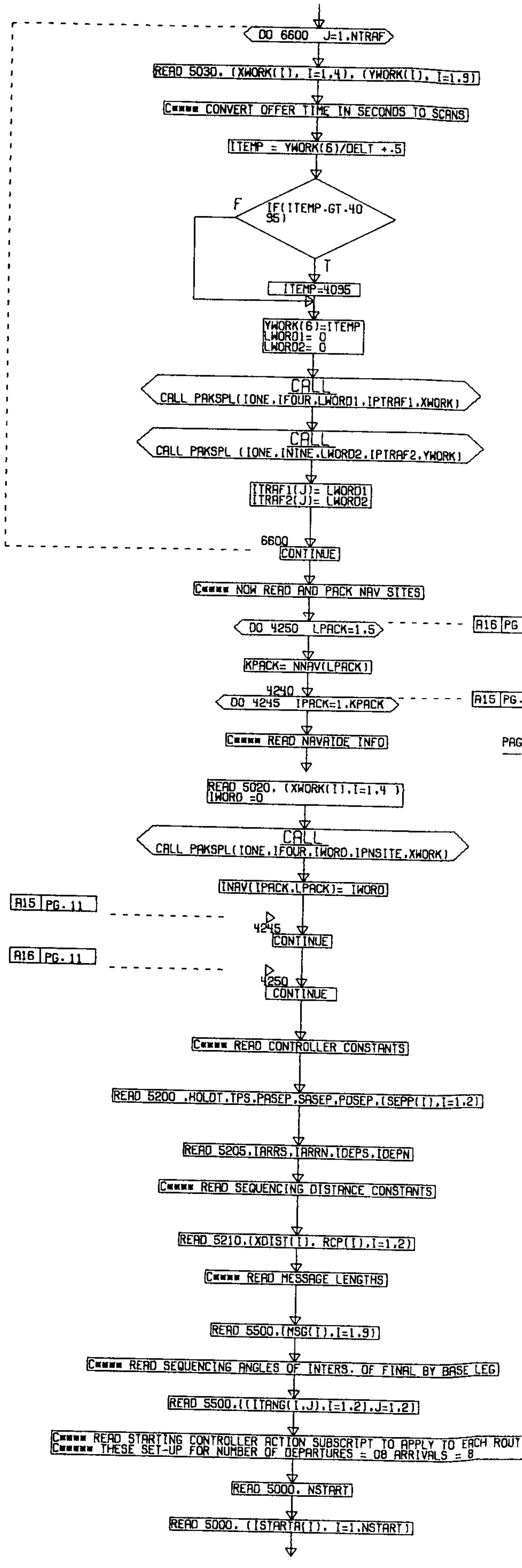
PAGE 7



CONTINUE
C***** READ PERFORMANCE TIMES
LENGTH= 6
C SCALE=1.0
READ 4000, (XWORK(I),I=1,NTYPE)
CALL
CALL PAKUNI(IPERFT,XWORK,NTYPE,LENGTH,SCALE)
C***** READ STANDARD TURN RATE
SCALE=10.0
C LENGTH=6
READ 4000, (XWORK(I),I=1,NTYPE)
CALL
CALL PAKUNI(IPFTR,XWORK,NTYPE,LENGTH,SCALE)
C****** IHEM(I) WHICH HEMISPHERE S CONTROLLER (N OF 0.0 IHEM=1,S OF 0.0
C***** IHEM = 2)
C***** ASSOCIATE ROUTES WITH HEMISPHERE
NROUTES= NDEPR+NARR
READ 5215, (IHEM(I),I=1, NROUTES)
C***** READ CONTROLLER ACTION ARRAY
READ 5000, NCACT
DO 6000 J=1,NCACT
A7 PG.8
READ 5600, (XWORK(I),I=1,6), (YWORK (I),I=1,2)
IWORD=0
PAGE 8
CALL
CALL PAKSPL(IONE,ISIX ,IWORD,IPCNTR1,XWORK)
ICNTRA1(J)= IWORD
IWORD = 0
CALL
CALL PAKSPL (IONE,ITWO,IWORD,IPCNTR2,YWORK)
ICNTRA2(J)= IWORD
A7 PG.7
6000
CONTINUE
C***** READ CHECK POINT ARRAY
READ 5000, NCHK
DO 6100 J=1,NCHK
READ 5605, (XWORK(I),I=1,6)
IWORD=0
CALL
CALL PAKSPL(IONE,ISIX,IWORD,IPCHKPT,XWORK)
ICHKPT(J)= IWORD
6100
CONTINUE
C***** READ IN IOPTN ARRAY (OPTION ARRAY)
READ 5000,NOPT
DO 6200 J=1,NOPT
READ 5610, (XWORK(I),I=1,9)
IWORD=0
CALL
CALL PAKSPL (IONE,ININE,IWORD,IPOPTN,XWORK)
IOPTN(J)=IWORD
6200
CONTINUE


C
C***** READ IN NNPATH ARRAY
READ 5000, NPTH
DO 6300 J=1,NPTH
READ 5610, (XWORK(I),I=1,7)
IWORD=0
CALL
CALL PAKSPL (IONE,ISEVEN, IWORD,IPNPATH,XWORK)
NNPATH(J)=IWORD
6300
CONTINUE
C
C***** READ IN SCHEDULES
READ 5000, NSKED
DO 6400 J=1,NSKED
A11
PG.10
READ 5615, (XWORK(I),I=1,8)
IF(XWORK(1).EQ .5.)
F
T
XWORK(3)=XWORK(3)*10.
IF(XWORK(6).EQ. 7. .OR.XWORK(6).EQ.10.)
F
T
XWORK(8)=XWORK(8)*10.
IF(XWORK(6).EQ .3.)
F
T
XWORK(7)=XWORK(7)*10.
IWORD=0
CALL
CALL PAKSPL (IONE,IEIGHT, IWORD,IPSKED,XWORK)
ISKED(J)= IWORD
A11
PG.9
6400
CONTINUE
C***** READ IN DISTANCES FOR SCHEDULES
READ 5620,(XWORK(I) ,I=1,NSKED)
SCALE = 10.
CALL
CALL PAKUNI(ISDST,XWORK,NSKED,ITEN,SCALE)
C***** READ IN IDROUTE (DIVERGENT ROUTE INFO)
READ 5000,NORTE
DO 6500 J=1,NORTE
READ 5611, (XWORK(I),I=1,5)
IWORD=0
CALL
CALL PAKSPL(IONE,IFIVE,IWORD,IPORT,XWORK)
IDROUTE (J)=IWORD
6500
CONTINUE
C***** READ MASTER TRAFFIC SAMPLE, (NTRAF WAS ON A HEADER CARD)
C PACK INTO ITRAF1 AND ITRAF2

PAGE 10


DO 6600 J=1,NTRAF
READ 5030, (XWORK(I), I=1,4), (YWORK(I), I=1,9)
C***** CONVERT OFFER TIME IN SECONDS TO SCANS
ITEMP = YWORK(6)/DELT +.5
IF(ITEMP.GT.40 95)
F
T
ITEMP=4095
YWORK(6)=ITEMP
LWORD1= 0
LWORD2= 0
CALL
CALL PAKSPL(IONE,IFOUR,LWORD1,IPTRAF1,XWORK)
CALL
CALL PAKSPL (IONE,ININE,LWORD2,IPTRAF2,YWORK)
ITRAF1(J)= LWORD1
ITRAF2(J)= LWORD2
6600
CONTINUE
C***** NOW READ AND PACK NAV SITES
DO 4250 LPACK=1,5
A16
PG
KPACK= NNAV(LPACK)
4240
DO 4245 IPACK=1,KPACK
A15
PG.
C***** READ NAVAIDE INFO
READ 5020, (XWORK(I),I=1,4)
IWORD =0
CALL
CALL PAKSPL(IONE,IFOUR,IWORD,IPNSITE,XWORK)
INAV(IPACK,LPACK)= IWORD
A15
PG. 11
4245
CONTINUE
A16
PG. 11
4250
CONTINUE
C***** READ CONTROLLER CONSTANTS
READ 5200 ,HOLDT,TPS,PASEP,SASEP,PDSEP,(SEPP(I),I=1,2)
READ 5205,IARRS,IARRN,IDEPS,IDEPN
C***** READ SEQUENCING DISTANCE CONSTANTS
READ 5210,(XDIST(I), RCP(I),I=1,2)
C***** READ MESSAGE LENGTHS
READ 5500,(MSG(I),I=1,9)
C***** READ SEQUENCING ANGLES OF INTERS. OF FINAL BY BASE LEG
READ 5500,((ITANG(I,J),I=1,2),J=1,2)
C***** READ STARTING CONTROLLER ACTION SUBSCRIPT TO APPLY TO EACH ROUT
C***** THESE SET-UP FOR NUMBER OF DEPARTURES = 08 ARRIVALS = 8
READ 5000, NSTART
READ 5000, (ISTARTA(I), I=1,NSTART)

```
END

C**** READ ALTOPT ARRAY I,J,K WHERE I IS NUMBER OF ALTITUDES
C****                         WHERE J IS TYPE OF ALTITUDES
C**** 6F10.0 FORMAT           WHERE K = 1 IS PROP A/C
C****                         WHERE K = 2 IS JET A/C
C**** NOTE ORDER ON CARDS

READ 5000,NALTOP ,JJ

DO 6700 K=1,2

DO 6700 I=1,NALTOP

READ 5212,(ALTOPT(I,J,K),J=1,JJ)

6700 CONTINUE

C**** READ ALTITUDE TO ASSOCIATE WITH SCHEDULES

READ 5000, NALTSKD

READ 5210,(ALTSKD(I),I=1,NALTSKD)

C**** READ SPEED VARIATION ARRAY K=1 FOR PROPS, K=2 FOR JETS

READ 5000, NSPDSTD

DO 6750 K=1,2

READ 5620,(SPDSTD(I,K),I=1,NSPDSTD))

6750 CONTINUE

C**** READ PATH NUMBER FOR STD HOLD, EMERG HOLD, AND POPUP

READ 5000, IHPATHS,IHPATHE,IPOPPA

C**** READ MISCELLANEOUS CONSTANTS

READ 5010,(IPOPR(I),IPOPL(I),I=1,2)

READ 5008,CKROLL,CKRC,CROLL,CRC,ZCON,HOCON
READ 5200,XDILS
READ 5200, GLIDE
READ 5008 ,CILS,CGLIDE,XLOCAL
```

PAGE 14

```
READ 5535,((RWYX(I,J),RWYY(I,J),I=1,4),RWYZ(J),J=1,2))

READ 5535,BETA

READ 5008,(XFINAL(I),I=1,2))

READ 5535,VECERR,BOUND

READ 5215,(IART(I),I=1,11)
READ 5210,(XA(I),YA(I),I=1,2)
READ 5210,(XC(I),YC(I),I=1,2)

READ 5502, ENROTA
READ 5502, ARRMNA, ARRMXA
READ 5503, SHOLDD, EHOLDD, HOLDSEP

READ 5535,(XRPOS(I),YRPOS(I),I=1,2)
READ 5515, (RAD(I), I=1,3), (SIG(I), I=1,3), IARG
SCALE =10.0
LENGTH=6
READ 4000,(XWORK(I),I=1,NTYPE)

CALL
CALL PAKUNI(IACFPER,XWORK,NTYPE,LENGTH,SCALE)

READ 5000,J

DO 6775 I=1,J

READ 5502,(XWORK(K),K=1,2)
LWORD = 0

CALL
CALL PAKSPL(IONE,ITWO,LWORD,IPETC,XWORK)

IETCOOR(I) = LWORD

6775 CONTINUE

SIG(1)=SIG(1)/6076.1
SIG(2)=SIG(2)*DTR

RETURN
```

```
SUBROUTINE INIT

LOGICAL LDISI, LDISO

COMMON / REALTIM/ ADC(32),DAC(64),LDISI(108),LDISO(196),
NOPER,NHOLD,NRESET,NTERM,NPRINT,NREAD

COMMON/TABLES/
     INTEG(80),      TABLE(075),      LOGIC(20)

LOGICAL LOGIC

*****
* * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN
*****

COMMON/MAJOR1/
     ACLS(50),        ALTOPT(3,7,2),  ALTSKD(9),
     ARRMNA,          ARRMXA,         BOUND,
     CGLIDE,          CILS,           CRA20,
     CRA30,           CRB20,          CRB30,
     DGRTD,           DTR,            EHOLOD,
     ENROTA(2),       FTNM,           G,
     GLIDE(2),        HOLDSEP,        IARRN,
     IARRS,           IART(11),       IBF(110),
     ICHKPT(40),      ICNTRA1(85),    ICNTRA2(85),
     IDELT,           IDEPN,          IDEPS,
     IDROUTE(10),     IHPATHE,        IHPATHS,
     INAV(10,5),      IOPTN(55),      IPAPATH(4,5),
     IPASKED(4,7),    IPCHKPT(4,6),   IPCNTR1(4,6),
     IPCNTR2(4,2),    IPDRT(4,5),     IPERFR(3,7),
     IPERFS(4,8),     IPERFT(3),      IPETA1(4,4),
     IPETA2(4,6),     IPFTR(3),       IPMSG2(4,6),
     IPMSG3(4,3),     IPNPATH(4,7),   IPNSITE(4,4),
     IPOPL(2),        IPOPPA,         IPOPR(2),
     IPOPTN(4,9),     IPSKED(4,8),    IPTAS(4,5)

COMMON/MAJOR2/
     IPTRAF1(4,4),    IPTRAF2(4,9),   ISDST(17),
     ISKDPU,          ISKED(86),      ISTARTA(21),
     ITANG(2,2),      ITRAF1(102),    ITRAF2(102),
     MAXK,            MF,
     NALTSKD,         NCP,            NDEPR,
     NNAV(5),         NNPATH(50),     NOUT(20),
     NSPOW,           OUT(26),        PI,
     PPMX(3),         PPMY(3),        RCP(2),
     RTD,             RWYX(4,2),      RWYY(4,2),
     RWYZ(2),         SHOLOD,         SPOSTD(9,2),
     SQX(6),          SQY(6),         TRIX(5),
     TRIY(5),         XA(2),          XC(2),
     XDILS(2),        XDIST(2),       XFINAL(2),
     XRPOS(2),        YA(2),          YC(2),
     YRPOS(2),        RAD(3),         SIG(3),        IARG
   , IETCOOR(36),     IPETC(4,2),     IACPPER(3)

*****

COMMON/TOES/ IONE,ITWO,ITHREE,IFOUR,IFIVE,ISIX,ISEVEN,IEIGHT,ININE
,ITEN
```

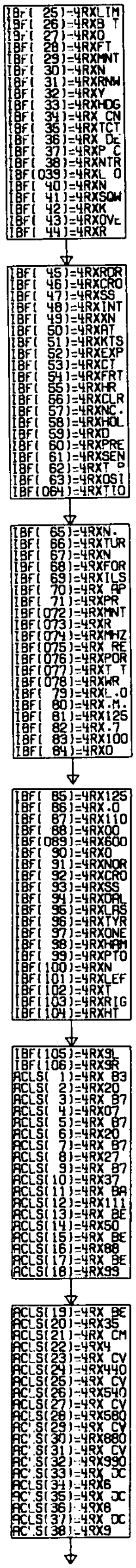
CROSSHAIR NO. 1


```
* * * TABLE,INTEG, AND LOGIC EQUIVALENCE FOR  SUBROUTINE INIT

EQUIVALENCE (INTEG(07), KRET)

EQUIVALENCE
     (INTEG(01),    ISCAL)     ,(INTEG(04),      ISCAN)
    ,(INTEG(09),     INCC)     ,(INTEG(11),         IO)
    ,(INTEG(12),     IOUT)     ,(INTEG(14),  IPLTAC(1))
    ,(INTEG(19),     IRUN)     ,(INTEG(20),       ISW1)
    ,(INTEG(23),     NTWO)     ,(INTEG(24),     IXYREC)
    ,(INTEG(30),    MAXAC)     ,(INTEG(33),       NCOM)
    ,(INTEG(34),  NCONTRL)     ,(INTEG(36),      NFRAM)
    ,(INTEG(38),    NTRAK)
    ,(INTEG(45),     ISTK)

EQUIVALENCE (INTEG(13), IOPT)
   ,  (INTEG(35), MAXQUE)

EQUIVALENCE
     (TABLE(13), CRTSCAL)
    ,(TABLE(18),     DELY)     ,(TABLE(20),       FREQ)
    ,(TABLE(21),        H)     ,(TABLE(27),      XOWNC)
    ,(TABLE(28),    YOWNC)     ,(TABLE(29),     UPERSC)
    ,(TABLE(30),   VPERSC)     ,(TABLE(40),       SCLA)
  , (TABLE(31), DELX1),          (TABLE(32), DELX2)
  , (TABLE(33), DELX3),          (TABLE(34), DELY1)
  , (TABLE(35), DELY2)
    ,(TABLE(41),     SCLE)     ,(TABLE(42),       SCLN)
    ,(TABLE(46), SQSCALE)      ,(TABLE(47),     VELCON)
  , (TABLE(49), XBOX),           (TABLE(50), YBOX)
    ,(TABLE(39), RADCP),         (TABLE(73), XSTP)
    ,(TABLE(74),     YSTP)     ,(TABLE(75),       VSTP)

EQUIVALENCE
     (LOGIC(01), CRTINIT)      ,(LOGIC(07),     WRTTPE)
    ,(LOGIC(08),  INITRO)      ,(LOGIC(09),     MSGTYP)
    ,(LOGIC(10), PERSPLT)      ,(LOGIC(11),     PLOTSW)
    ,(LOGIC(14),  TCONST)

* * * DIM. +  TYPE  FOR INIT

DIMENSION      IPLTAC(3)

LOGICAL     CRTINIT , WRTTPE , INITRO , MSGTYP , PERSPLT ,
            PLOTSW  , TCONST

**
```



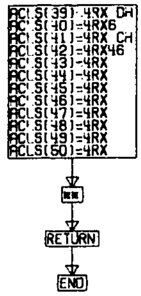
```
IONE=1
ITWO=2
ITHREE=3
IFOUR=4
IFIVE=5
ISIX=6
ISEVEN=7
IEIGHT=8
ININE=9
ITEN=10
INITRO=.TRUE.
MSGTYP=.FALSE.
PERSPLT=.T.
PLOTSW=.TRUE.
TCONST=.F.
CRTINIT=.TRUE.
WRTTPE=.FALSE.
LOGIC(20)=.FALSE.

**
* * * THE CRT SCALE (CRTSCAL) IS MEASURED IN CRT UNITS/ NAUTICAL MIL

CRTSCAL=12.5
DELX1 = 6.
DELX2 = -4.
DELX3 = -14.
DELY1 = 16.
DELY2 = 48.
DTR=.17453E-1
FREQ=0.
FTNM=6076.1
G=32.16

* * * H= THE TIME INTERVAL IN REAL TIME SECONDS/CYCLE

H=32./1024.
ISTK=0
INCC = 1
IO=0
IOPT=2
IOUT = 7
IPLTAC(1)=999
IPLTAC(2)=999
IPLTAC(3)=999
ISCAL=4
ISCAN=25
IRUN=0
ISW1=1
KRET = 4
NTWO=1
IXYREC=2
MAXAC=35
MAXQUE = 20
NFRAM=128
NTRAK=30

NCOM=36
NCONTRL=64

PI=3.14159

PID2=PI/2.

C**** PERSPECTIVE PLOT VARIABLES

UPERSC=0.12
VPERSC=0.12

C****

RADCP=2.5
RTD=57.2958
 SCLN=.0025
 SCLE=.0025
SCLA=.1/30000.
SQSCALE=1.0
VELCON = -2500.
XOWNC=.025
YOWNC=.025
XBOX = 24.
YBOX = 75.

**

CRA20=PID2-1.0-COS(20.*DTR)
CRB20=PID2-PI/9.+SIN(20.*DTR)
CRA30=PID2-1.0-COS(30.*DTR)
CRB30=PID2-PI/6.+SIN(30.*DTR)
PI20 = PI*20.
XKNFS=FTNM/3600.
XSTP=0.01
YSTP=0.01
VSTP=0.005

**

IBF(1)=ISETSYM(0.0.1)
IBF(2)=4RXOVE
IBF(3)=4RXR
IBF(4)=ISETSYM(0.0.0)
IBF(5)=ISETSYM(0.0.1)
IBF(6)=4RXHAN
IBF(7)=4RXD
IBF(8)=ISETSYM(0.0.0)
IBF(9)=ISETSYM(0.0.1)
IBF(10)=4RXEME
IBF(11)=4RXRG
IBF(12)=ISETSYM(0.0.0)
IBF(13)=ISETSYM(1.0.0)
IBF(14)=ISETSYM(2.0.0)
IBF(15)=ISETSYM(3.0.0)
IBF(020)=4RXCLR
IBF(021)=4RXD F
IBF( 22)=4RXOR
IBF( 23)=4RX .0
IBF( 24)=4RX. C
```

PAGE 5



PAGE 6

PAGE 7



```
OVERLAY (TAATMFL,2,0)
PROGRAM TATPRNT

COMMON/OVRBLK/ NTABLE,NINTEG,NLOGIC,NADC,NDAC,NLDISI,NLDISO,
    IJKBUFF(5),     IVARBUF(5),     KTYPE,          VARCHNG

EQUIVALENCE (INTEG(2),IPL),(INTEG(3),ISL)
LOGICAL LDISI, LDISO

COMMON / REALTIM/ ADC(32),DAC(64),LDISI(108),LDISO(196),
NOPER,NHOLD,NRESET,NTERM,NPRINT,NREAD

COMMON/TABLES/
    INTEG(80),      TABLE(075),     LOGIC(20)

LOGICAL LOGIC

*****
* * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN
*****

COMMON/MAJOR1/
    ACLS(50),       ALTOPI(3,7,2),  ALTSKD(9),
    ARRMNA,         ARRMXA,         BOUND,
    CGLIDE,         CILS,           CRA20,
    CRA30,          CRB20,          CRB30,
    DGRTD,          DIR,            EHOLDD,
    ENROTA(2),      FTNM,           G,
    GLIDE(2),       HOLDSEP,        IARRN,
    IARRS,          IART(11),       IBF(110),
    ICHKPT(40),     ICNTRA1(85),    ICNTRA2(85),
    IDELT,          IDEPN,          IDEPS,
    IDROUTE(10),    IHPATHE,        IHPATHS,
    INAV(10,5),     IOPTN(55),      IPAPATH(4,5),
    IPASKED(4,7),   IPCHKPT(4,6),   IPCNTR1(4,6),
    IPCNTR2(4,2),   IPORT(4,5),     IPERFR(3,7),
    IPERFS(4,8),    IPERFT(3),      IPETA1(4,4),
    IPETA2(4,6),    IPFTR(3),       IPMSG2(4,6),
    IPMSG3(4,3),    IPNPATH(4,7),   IPNSITE(4,4),
    IPOPL(2),       IPOPPA,         IPOPR(2),
    IPOPTN(4,9),    IPSKED(4,8),    IPTAS(4,5)

COMMON/MAJOR2/
    IPTRAF1(4,4),   IPTRAF2(4,9),   ISDST(17),
    ISKDPU,         ISKED(85),      ISTARTA(21),
    ITANG(2,2),     ITRAF1(102),    ITRAF2(102),
    MAXK,           MF,
    NALTSKD,        NCP,            NDEPR,
    NNAV(5),        NNPATH(50),     NOUT(20),
    NSPDW,          OUT(26),        PI,
    PPMX(3),        PPMY(3),        RCP(2),
    RTD,            RWYX(4,2),      RWYY(4,2),
    RWYZ(2),        SHOLDD,         SPDSTD(9,2),
    SQX(6),         SQY(6),         TRIX(5),
    TRIY(5),        XA(2),          XC(2),
    XDILS(2),       XDIST(2),       XFINAL(2),
    XRPOS(2),       YA(2),          YC(2),
    YRPOS(2),       RAD(3),         SIG(3),         IARG
  . IETCOOR(36),    IPETC(4,2),     IACFPER(3)
```

PAGE 2

CROSSHAIR NO. 1

```
*****
*****
* * * MISCELLANEOUS PRIMARY-TO-PRIMARY AND PRIMARY-TO-SECONDARY COMMON
*****      (0,0), (1,0), (2,0), (3,0), (4,0), (4,1), (4,2), (4,3)

COMMON/MISC/
    DPRIM,          IIHEM(21),      ITL,            NFRAMF,
    PITCH

*****

COMMON/TOES/ IONE,ITWO,ITHREE,IFOUR,IFIVE,ISIX,ISEVEN,IEIGHT,ININE
,ITEN

* * * TABLE INTEG LOGIC  EQIV. FOR OVERLAY(2,0)

EQUIVALENCE
    (INTEG(12),     IOUT)   ,(INTEG(19),     IRUN)
   ,(INTEG(40),     IT)     ,(INTEG(41),     NSCAN)
   ,(INTEG(68),     IDIAG)  ,(LOGIC(07),     WRTTPE)

* * * DIM. + TYPE FOR OVERLAY(2,0)

LOGICAL     WRTTPE

*****
* * * DIMENSION FOR OVERLAY(2,0)
*****

DIMENSION
    VEL(3)

*****
*****
* * * DIMENSIONS AND EQUIVALENCES FOR OVERLAY(2,0)
* * *

DIMENSION
    FSS(16)

EQUIVALENCE
    (LDISI(33),     FSS(1))

*****
* * * SET WHITE LIGHTS TO INDICATE OVERLAY COMBINATIONS IN EXECUTION.

LDISO(61)=LDISO(63)=LDISO(64)=LDISO(81)=LDISO(82)=LDISO(83)=.F.
LDISO(62)=.TRUE.

C***** PRINT CONTROL
```

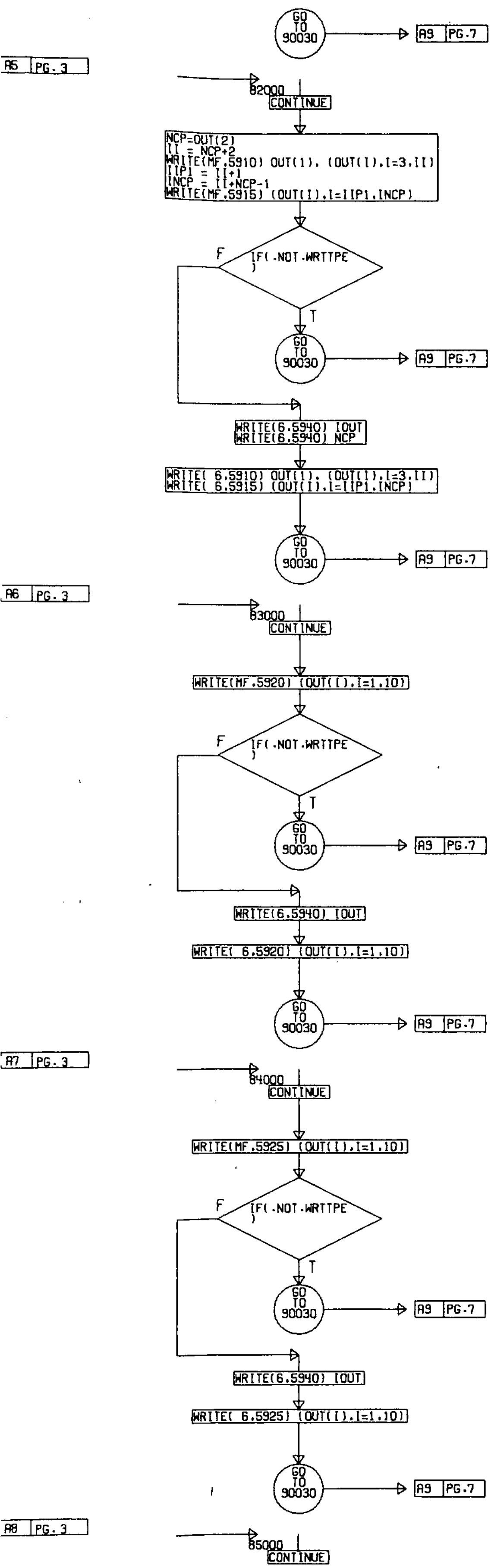


```
A1 | PG.7   ,7

90031
CALL
CALL PLAYBAK(900325)

IF(IT.EQ.ITL)   F / T
T: GO TO 80000

ITL=IT

WRITE(MF,5900) IT, NSCAN, IRUN

IF(.NOT.WRTTPE)   F / T
T: GO TO 80000

WRITE(6,5900) IT,NSCAN,IRUN

80000
CONTINUE

GO TO (81000, 82000, 83000, 84000, 85000, 86000, 87000, 88000)IOUT
```

A3	PG.4
A5	PG.5
A6	PG.5
A7	PG.6
A8	PG.6
A10	PG.7
A11	PG.7
A12	PG.7

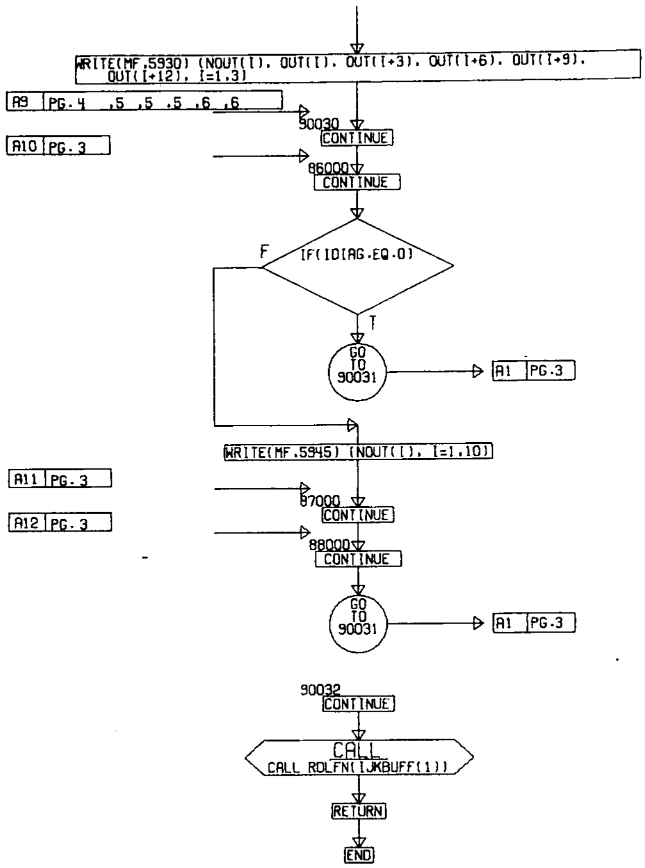
```
A3 | PG.3

81000
CONTINUE

WRITE(MF,5905) (OUT(I),I=1,6), OUT(9), OUT(11), (OUT(I),I=16,25)

IF(.NOT.WRTTPE)   F / T
T: GO TO 90030  -> A9 | PG.7

WRITE(6,5940) IOUT

DO 81500  I=13,15

81500
IF(OUT(I).GT.999.)   F / T
T: OUT(I)=999.

WRITE(6,5906) (OUT(I), I=1,11), (OUT(I), I=13,16),
    (OUT(I), I=18,20)
```

PAGE 5



PAGE 6

PAGE 7



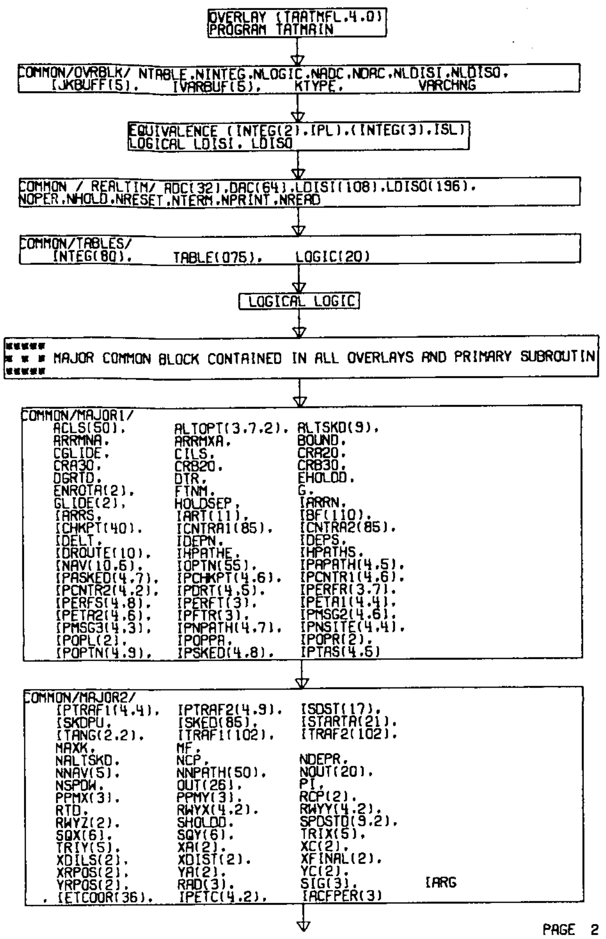
```
OVERLAY (TARTMFL,3,0)
PROGRAM TATREAD

COMMON/OVRBLK/ NTABLE,NINTEG,NLOGIC,NADC,NDAC,NLDISI,NLDISO,
      IJKBUFF(5),     IVARBUF(5),      KTYPE,          VARCHNG

EQUIVALENCE (INTEG(2),IPL),(INTEG(3),ISL)
LOGICAL LDISI, LDISO

COMMON / REALTIM/ ADC(32),DAC(64),LDISI(108),LDISO(196),
NOPER,NHOLD,NRESET,NTERM,NPRINT,NREAD

COMMON/TABLES/
    INTEG(80),       TABLE(075),      LOGIC(20)

LOGICAL LOGIC

*****
* * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN
*****

COMMON/MAJOR1/
     ACLS(50),          ALTOPT(3,7,2),   ALTSKD(9),
     ARRMNA,            ARRMXA,          BOUND,
     CGLIDE,            CILS,            CRA20,
     CRA30,             CRB20,           CRB30,
     DGRTD,             DTR,             EHOLDD,
     ENROTA(2),         FTNM,            G,
     GLIDE(2),          HOLDSEP,         IARRN,
     IARRS,             IART(11),        IBF(110),
     ICHKPT(40),        ICNTRA1(85),     ICNTRA2(85),
     IDELT,             IDEPN,           IDEPS,
     IDROUTE(10),       IHPATHE,         IHPATHS,
     INAV(10,5),        IOPTN(55),       IPAPATH(4,5),
     IPASKED(4,7),      IPCHKPT(4,6),    IPCNTR1(4,6),
     IPCNTR2(4,2),      IPDRT(4,5),      IPERFR(3,7),
     IPERFS(4,8),       IPERFT(3),       IPETA1(4,4),
     IPETA2(4,6),       IPFTR(3),        IPMSG2(4,6),
     IPMSG3(4,3),       IPNPATH(4,7),    IPNSITE(4,4),
     IPOPL(2),          IPOPPA,          IPOPR(2),
     IPOPTN(4,9),       IPSKED(4,8),     IPTAS(4,5)

COMMON/MAJOR2/
     IPTRAF1(4,4),      IPTRAF2(4,9),    ISDST(17),
     ISKDPU,            ISKED(85),       ISTARTA(21),
     ITANG(2,2),        ITRAF1(102),     ITRAF2(102),
     MAXK,              MF,
     NALTSKD,           NCP,             NDEPR,
     NNAV(5),           NNPATH(50),      NOUT(20),
     NSPDW,             OUT(26),         PI,
     PPMX(3),           PPMY(3),         RCP(2),
     RTD,               RWYX(4,2),       RWYY(4,2),
     RWYZ(2),           SHOLDD,          SPDSTD(9,2),
     SQX(6),            SQY(6),          TRIX(5),
     TRIY(5),           XA(2),           XC(2),
     XOILS(2),          XDIST(2),        XFINAL(2),
     XRPOS(2),          YA(2),           YC(2),
     YRPOS(2),          RAD(3),          SIG(3),        IARG
   , IETCOOR(36),       IPETC(4,2),      IACFPER(3)
```

CROSSHAIR NO. 1

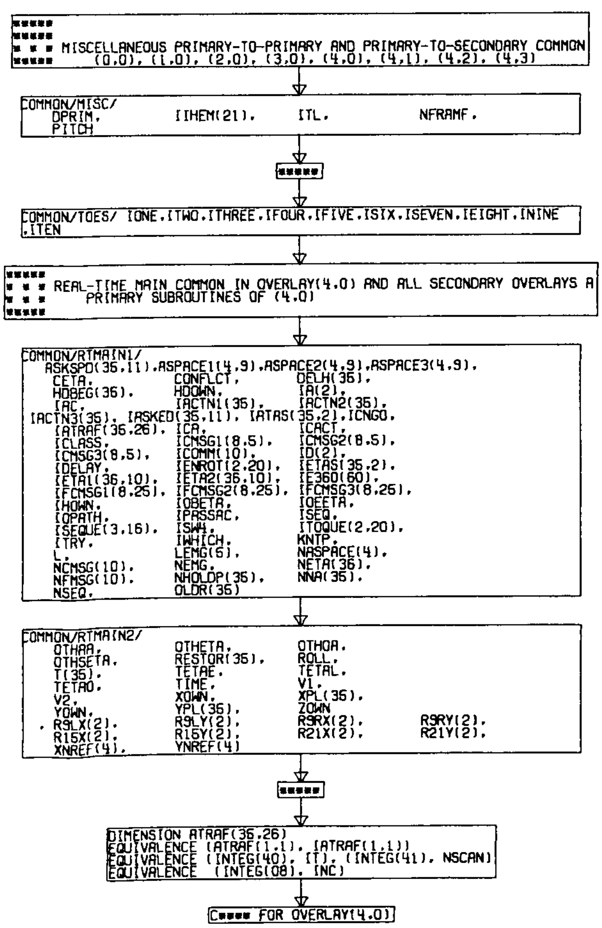

```
*****
*****
* * * MISCELLANEOUS PRIMARY-TO-PRIMARY AND PRIMARY-TO-SECONDARY COMMON
*****     (0,0), (1,0), (2,0), (3,0), (4,0), (4,1), (4,2), (4,3)

COMMON/MISC/
    OPRIM,          IITEM(21),      ITL,          NFRAME,
    PITCH

*****

COMMON/TOES/ IONE,ITWO,ITHREE,IFOUR,IFIVE,ISIX,ISEVEN,IEIGHT,ININE
,ITEN

* * * TABLE INTEG LOGIC EQIV. FOR OVERLAY(3,0)

EQUIVALENCE
    (INTEG(37),    NTRAF)     ,(TABLE(11),      TEMPS)
   ,(TABLE(16),    DELT)      ,(LOGIC(07),      WRTTPE)

* * * DIM + TYPE FOR OVERLAY(3,0)

LOGICAL     WRTTPE

*****
* * * DIMENSIONS AND EQUIVALENCES FOR OVERLAY(3,0)
*****

DIMENSION
    FSS(16),         IPERFS4(4),      XWORK(25),      YWORK(10)

EQUIVALENCE
    (IPERFS(1,4),    IPERFS4(1)),     (LDISI(33),    FSS(1))

*****
* * * SET WHITE LIGHTS TO INDICATE OVERLAY COMBINATIONS IN EXECUTION.

LDISO(61)=LDISO(62)=LDISO(64)=LDISO(81)=LDISO(82)=LDISO(83)=.F.
 LDISO(63)=.TRUE.

C***** READ CONTROL
C***** NOW READ AND PACK TRAF ARRAY XWORK INTO ITRAF(LPACK,1)
C                                   YWORK INTO ITRAF(LPACK,2)

READ 5000, NTRAF

C***** READ MASTER TRAFFIC SAMPLE.  (NTRAF WAS ON A HEADER CARD)
C      PACK INTO ITRAF1 AND ITRAF2

DO 6600  J=1,NTRAF  - - - - - - - A1 PG.3
```

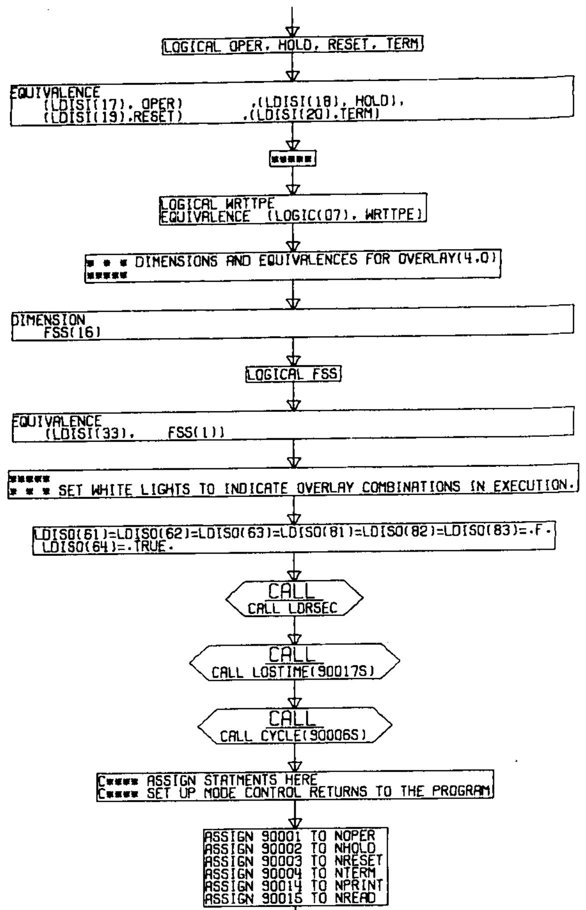


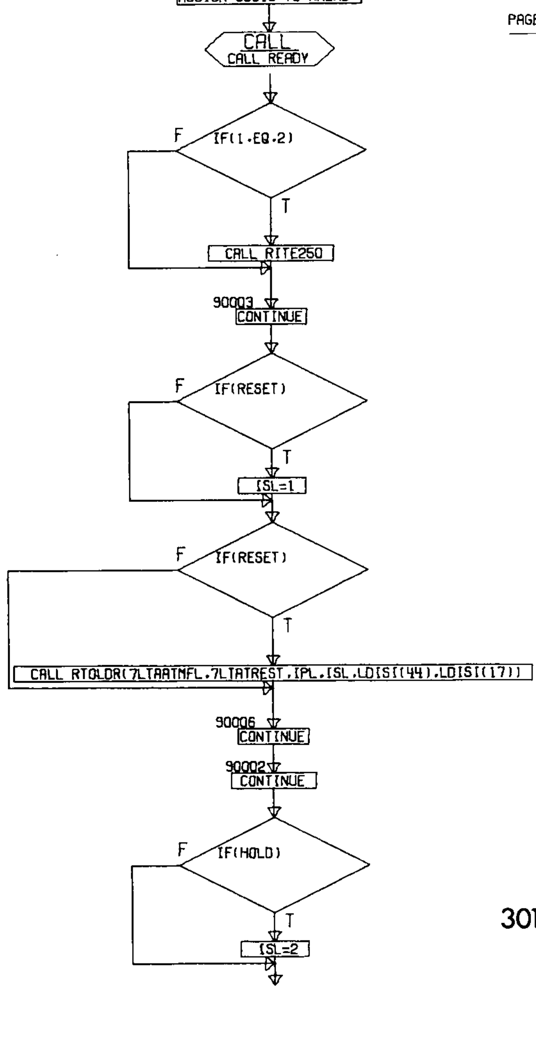
```
READ 5030, (XWORK(I), I=1,4), (YWORK(I), I=1,9)

C***** CONVERT OFFER TIME IN SECONDS TO SCANS

ITEMP = YWORK(6)/DELT +.5

IF(ITEMP.GT.40
95)
   F -> (skip)
   T -> ITEMP-4095

YWORK(6)=ITEMP
 WORD1= 0
 WORD2= 0

CALL
CALL PAKSPL(IONE,IFOUR,LWORD1,IPTRAF1,XWORK)

CALL
CALL PAKSPL (IONE,ININE,LWORD2,IPTRAF2,YWORK)

ITRAF1(J)= LWORD1
ITRAF2(J)= LWORD2

A1 PG. 2 - - - - - - - - - -
6600 CONTINUE

CALL
CALL RDLFN(IJKBUFF(1))

RETURN

END
```

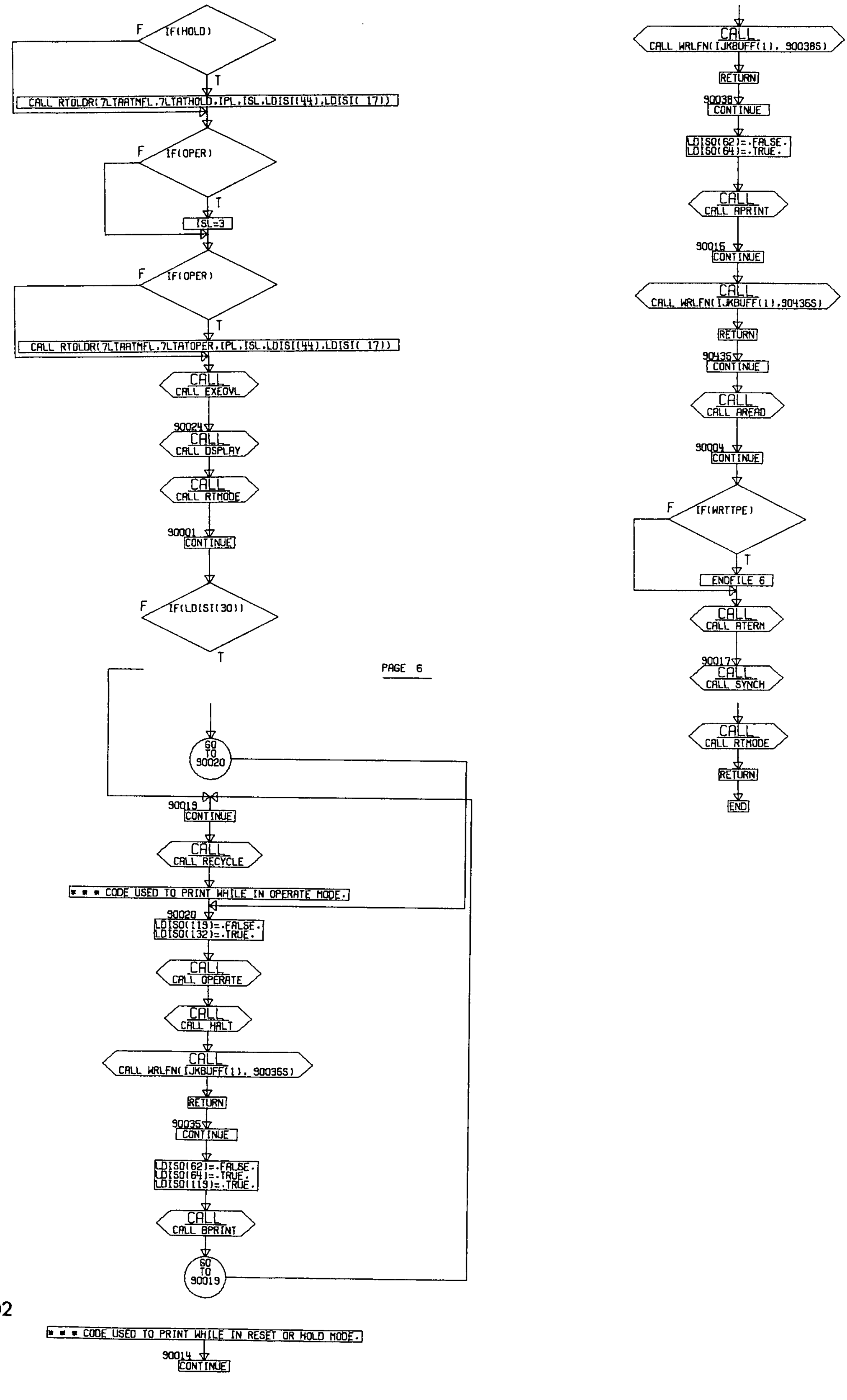

OVERLAY (TAATMFL.4.0)
PROGRAM TATMAIN
COMMON/OVRBLK/ NTABLE.NINTEG.NLOGIC.NADC.NDAC.NLDISI.NLDISO.
IJKBUFF(5). IVARBUF(5). KTYPE. VARCHNG
EQUIVALENCE (INTEG(2).IPL).(INTEG(3).ISL)
LOGICAL LDISI. LDISO
COMMON / REALTIM/ ADC(32).DAC(64).LDISI(108).LDISO(196).
NOPER.NHOLD.NRESET.NTERM.NPRINT.NREAD
COMMON/TABLES/
INTEG(80). TABLE(075). LOGIC(20)
LOGICAL LOGIC
*****
* * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN
*****
COMMON/MAJOR1/
ACLS(50). ALTOPT(3.7.2). ALTSKD(9).
ARRMNA. ARRMXA. BOUND.
CGLIDE. CILS. CRA20.
CRA30. CRB20. CRB30.
DGRTD. DTR. EHOLDD.
ENROTA(2). FTNM. G.
GLIDE(2). HOLDSEP. IARRN.
IARRS. IART(11). IBF(110).
ICHKPT(40). ICNTRA1(85). ICNTRA2(85).
IDELT. IDEPN. IDEPS.
IDROUTE(10). IHPATHE. IHPATHS.
INAV(10.6). IOPTN(55). IPAPATH(4.5).
IPASKED(4.7). IPCHKPT(4.6). IPCNTR1(4.6).
IPCNTR2(4.2). IPDRT(4.5). IPERFR(3.7).
IPERFS(4.8). IPERFT(3). IPETA1(4.4).
IPETA2(4.6). IPFTR(3). IPMSG2(4.6).
IPMSG3(4.3). IPNPATH(4.7). IPNSITE(4.4).
IPOPL(2). IPOPPA. IPOPR(2).
IPOPTN(4.9). IPSKED(4.8). IPTAS(4.6)
COMMON/MAJOR2/
IPTRAF1(4.4). IPTRAF2(4.9). ISDST(17).
ISKDPU. ISKED(85). ISTARTA(21).
ITANG(2.2). ITRAF1(102). ITRAF2(102).
MAXK. MF.
NALTSKD. NCP. NDEPR.
NNAV(5). NNPATH(50). NOUT(20).
NSPDW. OUT(26). PI.
PPMX(3). PPMY(3). RCP(2).
RTD. RWYX(4.2). RWYY(4.2).
RWYZ(2). SHOLDD. SPDSTD(9.2).
SQX(6). SQY(6). TRIX(5).
TRIY(5). XA(2). XC(2).
XDILS(2). XDIST(2). XFINAL(2).
XRPOS(2). YA(2). YC(2).
YRPOS(2). RAD(3). SIG(3). IARG
. IETCOOR(36). IPETC(4.2). IACFPER(3)

SHAIR NO. 1


*****
*****
* * * MISCELLANEOUS PRIMARY-TO-PRIMARY AND PRIMARY-TO-SECONDARY COMMON
***** (0.0). (1.0). (2.0). (3.0). (4.0). (4.1). (4.2). (4.3)
COMMON/MISC/
DPRIM. IIHEM(21). ITL. NFRAMF.
PITCH
*****
COMMON/TOES/ IONE.ITWO.ITHREE.IFOUR.IFIVE.ISIX.ISEVEN.IEIGHT.ININE
.ITEN
*****
* * * REAL-TIME MAIN COMMON IN OVERLAY(4.0) AND ALL SECONDARY OVERLAYS A
* * * PRIMARY SUBROUTINES OF (4.0)
*****
COMMON/RTMAIN1/
ASKSPD(35.11).ASPACE1(4.9).ASPACE2(4.9).ASPACE3(4.9).
CETA. CONFLCT. DELH(35).
HDBEG(36). HDOWN. IA(2).
IAC. IACTN1(35). IACTN2(35).
IACTN3(35). IASKED(35.11). IATAS(35.2).ICNGO.
IATRAF(35.26). ICA. ICACT.
ICLASS. ICMSG1(8.5). ICMSG2(8.5).
ICMSG3(8.5). ICOMM(10). ID(2).
IDELAY. IENROT(2.20). IETAS(35.2).
IETA1(36.10). IETA2(36.10). IE360(60).
IFCMSG1(8.25). IFCMSG2(8.25). IFCMSG3(8.25).
IHOWN. IOBETA. IOEETA.
IOPATH. IPASSAC. ISEQ.
ISEQUE(3.16). ISWH. ITOQUE(2.20).
ITRY. IWHICH. KNTP.
L. LEMG(5). NASPACE(4).
NCMSG(10). NEMG. NETA(36).
NFMSG(10). NHOLDP(35). NNA(35).
NSEQ. OLDR(35)
COMMON/RTMAIN2/
OTHAA. OTHETA. OTHOA.
OTHSETA. RESTOR(35). ROLL.
T(35). TETAE. TETAL.
TETAO. TIME. V1.
V2. XOWN. XPL(35).
YOWN. YPL(35). ZOWN
. R9LX(2). R9LY(2). R9RX(2). R9RY(2).
R15X(2). R15Y(2). R21X(2). R21Y(2).
XNREF(4). YNREF(4)
*****
DIMENSION ATRAF(35.26)
EQUIVALENCE (ATRAF(1.1). IATRAF(1.1))
EQUIVALENCE (INTEG(40). IT). (INTEG(41). NSCAN)
EQUIVALENCE (INTEG(08). INC)
C**** FOR OVERLAY(4.0)


LOGICAL OPER. HOLD. RESET. TERM
EQUIVALENCE
(LDISI(17). OPER) .(LDISI(18). HOLD).
(LDISI(19).RESET) .(LDISI(20).TERM)
*****
LOGICAL WRTTPE
EQUIVALENCE (LOGIC(07). WRTTPE)
* * * DIMENSIONS AND EQUIVALENCES FOR OVERLAY(4.0)
*****
DIMENSION
FSS(16)
LOGICAL FSS
EQUIVALENCE
(LDISI(33). FSS(1))
*****
* * * SET WHITE LIGHTS TO INDICATE OVERLAY COMBINATIONS IN EXECUTION.
LDISO(61)=LDISO(62)=LDISO(63)=LDISO(81)=LDISO(82)=LDISO(83)=.F.
LDISO(64)=.TRUE.
CALL
CALL LDRSEC
CALL
CALL LOSTIME(90017S)
CALL
CALL CYCLE(90006S)
C**** ASSIGN STATMENTS HERE
C**** SET UP MODE CONTROL RETURNS TO THE PROGRAM
ASSIGN 90001 TO NOPER
ASSIGN 90002 TO NHOLD
ASSIGN 90003 TO NRESET
ASSIGN 90004 TO NTERM
ASSIGN 90014 TO NPRINT
ASSIGN 90015 TO NREAD


CALL
CALL READY
F
IF(1.EQ.2)
T
CALL RITE250
90003
CONTINUE
F
IF(RESET)
T
ISL=1
F
IF(RESET)
T
CALL RTOLDR(7LTAATMFL.7LTATREST.IPL.ISL.LDISI(44).LDISI(17))
90006
CONTINUE
90002
CONTINUE
F
IF(HOLD)
T
ISL=2


IF(HOLD)
F
T
CALL RTOLDR(7LTAATMFL,7LTATHOLD,IPL,ISL,LDISI(44),LDISI(17))
IF(OPER)
F
T
ISL=3
IF(OPER)
F
T
CALL RTOLDR(7LTAATMFL,7LTATOPER,IPL,ISL,LDISI(44),LDISI(17))
CALL
CALL EXEOVL
90024
CALL
CALL DSPLAY
CALL
CALL RTMODE
90001
CONTINUE
IF(LDISI(30))
F
T
PAGE 6
GO TO 90020
90019
CONTINUE
CALL
CALL RECYCLE
* * * CODE USED TO PRINT WHILE IN OPERATE MODE.
90020
LDISO(119)=.FALSE.
LDISO(132)=.TRUE.
CALL
CALL OPERATE
CALL
CALL HALT
CALL
CALL WRLFN(IJKBUFF(1), 90035S)
RETURN
90035
CONTINUE
LDISO(62)=.FALSE.
LDISO(64)=.TRUE.
LDISO(119)=.TRUE.
CALL
CALL BPRINT
GO TO 90019
* * * CODE USED TO PRINT WHILE IN RESET OR HOLD MODE.
90014
CONTINUE
CALL
CALL WRLFN(IJKBUFF(1), 90038S)
RETURN
90038
CONTINUE
LDISO(62)=.FALSE.
LDISO(64)=.TRUE.
CALL
CALL APRINT
90016
CONTINUE
CALL
CALL WRLFN(IJKBUFF(1),90436S)
RETURN
90436
CONTINUE
CALL
CALL AREAD
90004
CONTINUE
IF(WRTTPE)
F
T
ENDFILE 6
CALL
CALL ATERM
90017
CALL
CALL SYNCH
CALL
CALL RTMODE
RETURN
END
PAGE

```
OVERLAY (TRATMFL.4.1)
PROGRAM  TATREST
```

```
COMMON/OVRBLK/ NTABLE.NINTEG.NLOGIC.NADC.NDAC.NLDISI.NLDISO.
     IJKBUFF(5).      IVARBUF(5).      KTYPE.           VARCHNG
```

```
EQUIVALENCE (INTEG(2).IPL).(INTEG(3).ISL)
LOGICAL LDISI. LDISO
```

```
COMMON / REALTIM/ ADC(32).DAC(64).LDISI(108).LDISO(196).
NOPER.NHOLD.NRESET.NTERM.NPRINT.NREAD
```

```
COMMON/TABLES/
   INTEG(80).      TABLE(075).      LOGIC(20)
```

```
LOGICAL LOGIC
```

```
*****
* * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN
*****
```

```
COMMON/MAJOR1/
     ACLS(50).        ALTOPT(3.7.2). ALTSKD(9).
     ARRMNA.          ARRMXA.        BOUND.
     CGLIDE.          CILS.          CRA20.
     CRA30.           CRB20.         CRB30.
     DGRTD.           DTR.           EHOLDD.
     ENROTA(2).       FTNM.          G.
     GLIDE(2).        HOLDSEP.       IARRN.
     IARRS.           IART(11).      IBF(110).
     ICHKPT(40).      ICNTRA1(85).   ICNTRA2(85).
     IDELT.           IDEPN.         IDEPS.
     IDROUTE(10).     IHPATHE.       IHPATHS.
     INAV(10.5).      IOPTN(55).     IPAPATH(4.5).
     IPASKED(4.7).    IPCHKPT(4.6).  IPCNTR1(4.6).
     IPCNTR2(4.2).    IPORT(4.5).    IPERFR(3.7).
     IPERFS(4.8).     IPERFT(3).     IPETA1(4.4).
     IPETA2(4.6).     IPFTR(3).      IPMSG2(4.6).
     IPMSG3(4.3).     IPNPATH(4.7).  IPNSITE(4.4).
     IPOPL(2).        IPOPPA.        IPOPR(2).
     IPOPTN(4.9).     IPSKED(4.8).   IPTAS(4.5)
```

```
COMMON/MAJOR2/
     IPTRAF1(4.4).    IPTRAF2(4.9).  ISDST(17).
     ISKDPU.          ISKED(85).     ISTARTA(21).
     ITANG(2.2).      ITRAF1(102).   ITRAF2(102).
     MAXK.            MF.
     NALTSKD.         NCP.           NDEPR.
     NNAV(5).         NNPATH(50).    NOUT(20).
     NSPDW.           OUT(26).       PI.
     PPMX(3).         PPMY(3).       RCP(2).
     RTD.             RWYX(4.2).     RWYY(4.2).
     RWYZ(2).         SHOLDD.        SPOSTD(9.2).
     SQX(6).          SQY(6).        TRIX(6).
     TRIY(5).         XA(2).         XC(2).
     XDILS(2).        XDIST(2).      XFINAL(2).
     XRPOS(2).        YA(2).         YC(2).
     YRPOS(2).        RAD(3).        SIG(3).        IARG
   . IETCOOR(36).     IPETC(4.2).    IACFPER(3)
```


SHAIR NO. 1

```
*****
*****
* * * MISCELLANEOUS PRIMARY-TO-PRIMARY AND PRIMARY-TO-SECONDARY COMMON
*****       (0.0). (1.0). (2.0). (3.0). (4.0). (4.1). (4.2). (4.3)
```

```
COMMON/MISC/
    DPRIM.          IIHEM(21).      ITL.            NFRAMF.
    PITCH
```

```
*****
```

```
COMMON/TOES/ IONE.ITWO.ITHREE.IFOUR.IFIVE.ISIX.ISEVEN.IEIGHT.ININE
.ITEN
```

```
*****
* * * REAL-TIME MAIN COMMON IN OVERLAY(4.0) AND ALL SECONDARY OVERLAYS A
* * *      PRIMARY SUBROUTINES OF (4.0)
*****
```

```
COMMON/RTMAIN1/
   ASKSPD(35.11).ASPACE1(4.9).ASPACE2(4.9).ASPACE3(4.9).
    CETA.            CONFLCT.        DELH(35).
    HDBEG(35).       HDOWN.          IA(2).
    IAC.             IACTN1(35).     IACTN2(35).
 IACTN3(35). IASKED(35.11). IATAS(35.2).ICNGO.
    IATRAF(35.26).   ICA.            ICACT.
    ICLASS.          ICMSG1(8.5).    ICMSG2(8.5).
    ICMSG3(8.5).     ICOMM(10).      ID(2).
    IDELAY.          IENROT(2.20).   IETAS(35.2).
    IETA1(36.10).    IETA2(36.10).   IE360(60).
    IFCMSG1(8.25).   IFCMSG2(8.25).  IFCMSG3(8.25).
    IHOWN.           IOBETA.         IOEETA.
    IOPATH.          IPASSAC.        ISEQ.
    ISEQUE(3.16).    ISW4.           ITOQUE(2.20).
    ITRY.            IWHICH.         KNTP.
    L.               LEMG(6).        NASPACE(4).
    NCMSG(10).       NEMG.           NETA(36).
    NFMSG(10).       NHOLDP(35).     NNA(35).
    NSEQ.            OLDR(35)
```

```
COMMON/RTMAIN2/
    DTHAA.           DTHETA.         DTHOA.
    DTHSETA.         RESTOR(35).     ROLL.
    T(35).           TETAE.          TETAL.
    TETAO.           TIME.           V1.
    V2.              XOWN.           XPL(35).
    YOWN.            YPL(35).        ZOWN
  . R9LX(2).         R9LY(2).        R9RX(2).        R9RY(2).
    R15X(2).         R15Y(2).        R21X(2).        R21Y(2).
    XNREF(4).        YNREF(4)
```

```
*****
* * * DIM. + TYPE FOR OVERLAY(4.1)
```

```
DIMENSION    IATSPL(4.1)
LOGICAL    INITRO . TCONST
```

```
* * * TABLE INTEG LOGIC EQIV. FOR OVERLAY(4.1)
```



```
EQUIVALENCE
    (INTEG(25). IDREL).
    (TABLE(07). XREL).                    (TABLE(08). YREL).
    (TABLE(09). ZREL).                    (TABLE(10). PSIREL)
```

```
EQUIVALENCE
    (INTEG(08).        INC)      .(INTEG(10).          INUM)
   .(INTEG(17).         IR)      .(INTEG(19).          IRUN)
   .(INTEG(21).       ISW2)      .(INTEG(22).          ISW3)
   .(INTEG(30). MAXAC)
   .(INTEG(40).         IT)      .(INTEG(41).         NSCAN)
   .(INTEG(68). IDIAG)
   .(INTEG(70).       IBEG)      .(INTEG(71).IATSPL(1.1))
   .(TABLE(11).      TEMPS)      .(TABLE(26).BETA).(TABLE(45).RVECT)
   .(LOGIC(08).     INITRC)      .(LOGIC(14).       TCONST)
```

```
*****
* * * DIMENSIONS AND EQUIVALENCES FOR OVERLAY(4.1)
*****
```

```
DIMENSION
    FSS(16)
```

```
EQUIVALENCE
    (LDISI(33).      FSS(1))
```

```
*****
* * * SET WHITE LIGHTS TO INDICATE OVERLAY COMBINATIONS IN EXECUTION.
```

```
LDISO(82)=LDISO(83)=.FALSE.
LDISO(81)=.TRUE.
```

```
*****************************************************************
C**** INITIALIZATION - RESET *************************************
*****************************************************************
```

```
90003  CONTINUE
```

```
KNTP=0
```

```
IF(.NOT.INITRO
)
```

T

F

```
GO
TO
100
```

A13 PG.7

```
CALL
CALL HALT
```



```
C**** INITIALIZATION OF COUNTERS AND ZEROING OUT OF ARRAYS
```

```
CETA=0.
CONFLCT=0.
IARG = 9
ICA=0
ICACT=0
```

```
DO 1000   I=1.60
```

```
IE360(I)=0
```

```
1000  CONTINUE
```

```
IPASSAC=0
ITRY=0
IWHICH=0
```

```
DO 1010   I=1.5
```

```
LEMG(I)=0
```

```
1010  CONTINUE
```

```
DO 1020   I=1.4
```

```
NASPACE(I)=0
```

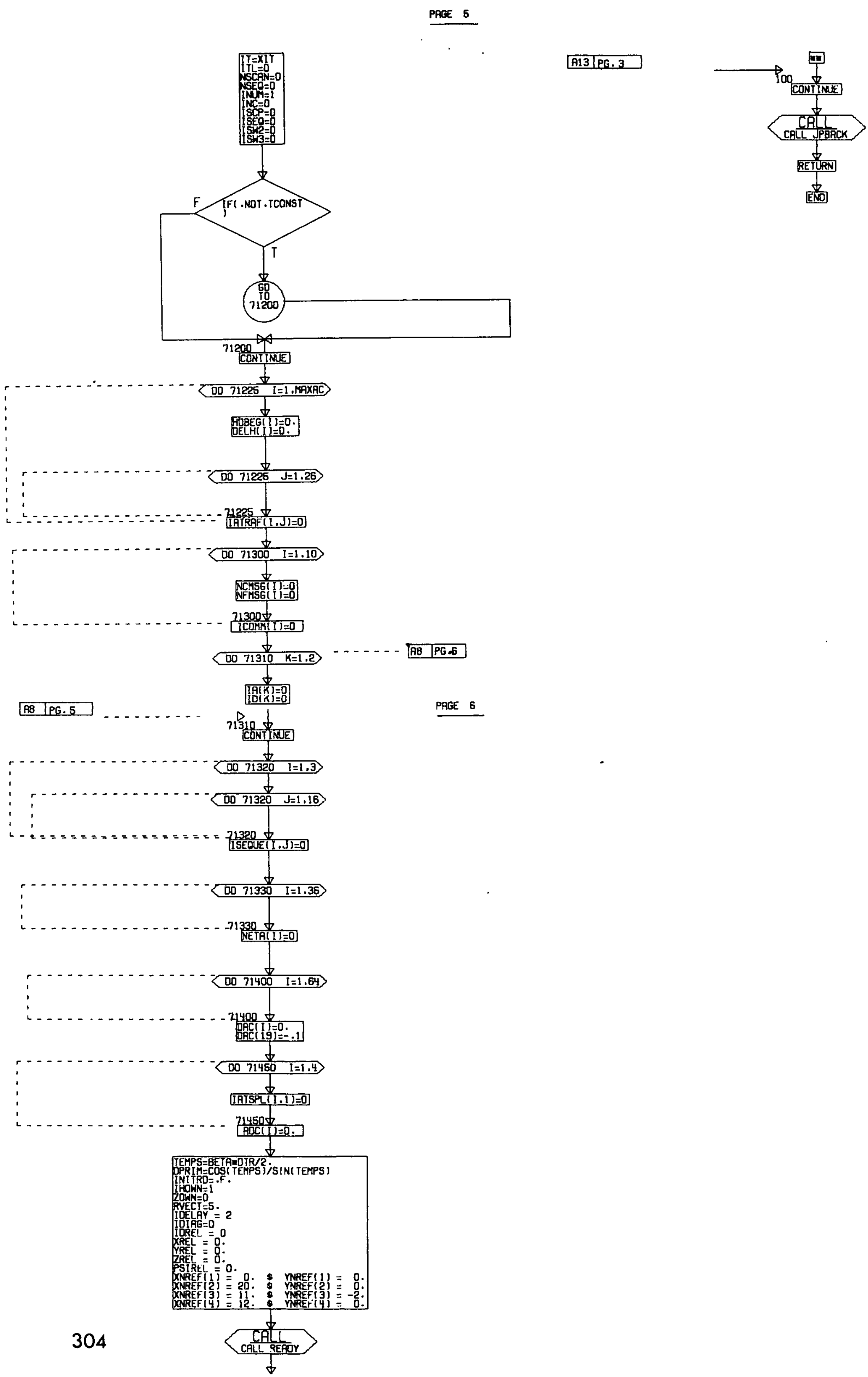
```
1020  CONTINUE
```

```
NEMG=0
```

```
C**** INITIALIZE COUNTERS
C**** IT - SCAN COUNT
C**** NSCAN - NUMBER OF AIRCRAFT IN SCAN
C**** INUM - WHICH AIRCRAFT ENTERS THE PATTERN NEXT FROM MASTER TRAF SAM
```

```
IBEG=0
IR=3
IRUN=IRUN+1
```

```
CALL
CALL UPKSPL(ISIX.ISIX.ITRAF2(1).IPTRAF2.XIT)
```


PAGE 5
IT=XIT
ITL=0
NSCAN=0
NSEQ=0
INUM=1
INC=0
ISCP=0
ISEQ=0
ISW2=0
ISW3=0
IF(.NOT.TCONST)
F
T
GO TO 71200
71200
CONTINUE
DO 71225 I=1,MAXAC
HDBEG(I)=0.
DELH(I)=0.
DO 71225 J=1,26
71225
IATRAF(I,J)=0
DO 71300 I=1,10
NCMSG(I)=0
NFMSG(I)=0
71300
ICOMM(I)=0
DO 71310 K=1,2
A8 PG.6
IA(K)=0
ID(K)=0
A8 PG.5
PAGE 6
71310
CONTINUE
DO 71320 I=1,3
DO 71320 J=1,16
71320
ISEQUE(I,J)=0
DO 71330 I=1,36
71330
NETA(I)=0
DO 71400 I=1,64
71400
DAC(I)=0.
DAC(19)=-.1
DO 71450 I=1,4
IATSPL(I,1)=0
71450
ADC(I)=0.
TEMPS=BETA*DTR/2.
DPRIM=COS(TEMPS)/SIN(TEMPS)
INITRD=.F.
IHOWN=1
ZOWN=0
RVECT=5.
IDELAY = 2
IDIAG=0
IDREL = 0
XREL = 0.
YREL = 0.
ZREL = 0.
PSIREL = 0.
XNREF(1) = 0. $ YNREF(1) = 0.
XNREF(2) = 20. $ YNREF(2) = 0.
XNREF(3) = 11. $ YNREF(3) = -2.
XNREF(4) = 12. $ YNREF(4) = 0.
CALL
CALL READY
A13 PG.3
100
CONTINUE
CALL
CALL JPBACK
RETURN
END

```
OVERLAY (TAATMFL,4,2)
PROGRAM TATHOLD

COMMON/OVRBLK/ NTABLE,NINTEG,NLOGIC,NADC,NDAC,NLDISI,NLDISO,
    IJKBUFF(5),      IVARBUF(5),     KTYPE,         VARCHNG

EQUIVALENCE (INTEG(2),IPL),(INTEG(3),ISL)
LOGICAL LDISI, LDISO

COMMON / REALTIM/ ADC(32),DAC(64),LDISI(108),LDISO(196),
NOPER,NHOLD,NRESET,NTERM,NPRINT,NREAD

COMMON/TABLES/
    INTEG(80),      TABLE(075),    LOGIC(20)

LOGICAL LOGIC

*****
* * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN
*****

COMMON/MAJOR1/
    ACLS(50),       ALTOPT(3,7,2), ALTSKD(9),
    ARRMNA,         ARRMXA,        BOUND,
    CGLIDE,         CILS,          CRA20,
    CRA30,          CRB20,         CRB30,
    DGRTD,          DTR,           EHOLDD,
    ENROTA(2),      FTNM,          G,
    GLIDE(2),       HOLDSEP,       IARRN,
    IARRS,          IART(11),      IBF(110),
    ICHKPT(40),     ICNTRA1(85),   ICNTRA2(85),
    IDELT,          IDEPN,         IDEPS,
    IDROUTE(10),    IHPATHE,       IHPATHS,
    INAV(10,5),     IOPTN(55),     IPAPATH(4,5),
    IPASKED(4,7),   IPCHKPT(4,6),  IPCNTR1(4,6),
    IPCNTR2(4,2),   IPDRT(4,5),    IPERFR(3,7),
    IPERFS(4,8),    IPERFT(3),     IPETA1(4,4),
    IPETA2(4,6),    IPFTR(3),      IPMSG2(4,6),
    IPMSG3(4,3),    IPNPATH(4,7),  IPNSITE(4,4),
    IPOPL(2),       IPOPPA,        IPOPR(2),
    IPOPTN(4,9),    IPSKED(4,8),   IPTAS(4,5)

COMMON/MAJOR2/
    IPTRAF1(4,4),   IPTRAF2(4,9),  ISDST(17),
    ISKDPU,         ISKED(85),     ISTARTA(21),
    ITANG(2,2),     ITRAF1(102),   ITRAF2(102),
    MAXK,           MF,
    NALTSKD,        NCP,           NDEPR,
    NNAV(5),        NNPATH(50),    NOUT(20),
    NSPDW,          OUT(26),       PI,
    PPMX(3),        PPMY(3),       RCP(2),
    RTD,            RWYX(4,2),     RWYY(4,2),
    RWYZ(2),        SHOLDD,        SPDSTD(9,2),
    SQX(6),         SQY(6),        TRIX(5),
    TRIY(5),        XA(2),         XC(2),
    XDILS(2),       XDIST(2),      XFINAL(2),
    XRPOS(2),       YA(2),         YC(2),
    YRPOS(2),       RAD(3),        SIG(3),         IARG
  , IETCOOR(36),    IPETC(4,2),    IACFPER(3)
```


IR NO. 1

```
*****
*****
* * * MISCELLANEOUS PRIMARY-TO-PRIMARY AND PRIMARY-TO-SECONDARY COMMON
*****     (0,0), (1,0), (2,0), (3,0), (4,0), (4,1), (4,2), (4,3)

COMMON/MISC/
    DPRIM,          IIHEM(21),     ITL,            NFRAMF,
    PITCH

*****

COMMON/TOES/ IONE,ITWO,ITHREE,IFOUR,IFIVE,ISIX,ISEVEN,IEIGHT,ININE
,ITEN

*****
* * * REAL-TIME MAIN COMMON IN OVERLAY(4,0) AND ALL SECONDARY OVERLAYS A
* * *     PRIMARY SUBROUTINES OF (4,0)
*****

COMMON/RTMAIN1/
   ASKSPD(35,11),ASPACE1(4,9),ASPACE2(4,9),ASPACE3(4,9),
    CETA,           CONFLCT,       DELH(35),
    HDBEG(35),      HDOWN,         IA(2),
    IAC,            IACTN1(35),    IACTN2(35),
 IACTN3(35), IASKED(35,11), IATAS(35,2),ICNGO,
    IATRAF(35,26),  ICA,           ICACT,
    ICLASS,         ICMSG1(8,5),   ICMSG2(8,5),
    ICMSG3(8,5),    ICOMM(10),     ID(2),
    IDELAY,         IENROT(2,20),  IETAS(35,2),
    IETA1(36,10),   IETA2(36,10),  IE360(60),
    IFCMSG1(8,25),  IFCMSG2(8,25), IFCMSG3(8,25),
    IHOWN,          IOBETA,        IOEETA,
    IOPATH,         IPASSAC,       ISEQ,
    ISEQUE(3,16),   ISW4,          ITOQUE(2,20),
    ITRY,           IWHICH,        KNTP,
    L,              LEMG(5),       NASPACE(4),
    NCMSG(10),      NEMG,          NETA(36),
    NFMSG(10),      NHOLDP(35),    NNA(35),
    NSEQ,           OLDR(35)

COMMON/RTMAIN2/
    OTHAA,          OTHETA,        OTHOA,
    OTHSETA,        RESTOR(35),    ROLL,
    T(35),          TETAE,         TETAL,
    TETAO,          TIME,          V1,
    V2,             XOWN,          XPL(35),
    YOWN,           YPL(35),       ZOWN
  , R9LX(2),        R9LY(2),       R9RX(2),        R9RY(2),
    R15X(2),        R15Y(2),       R21X(2),        R21Y(2),
    XNREF(4),       YNREF(4)

*****
* * * DIM. + TYPE FOR OVERLAY(4,2)

DIMENSION      IATSPL(4,1)
LOGICAL        CRTINIT , PERSPLT , PLOTSW

* * * TABLE, INTEG, AND LOGIC FOR  OVERLAY (4,2)
```



```
EQUIVALENCE
    (INTEG(01),    ISCAL)     ,(INTEG(19),       IRUN)
   ,(INTEG(20),     ISW1)     ,(INTEG(23),       NTWO)
   ,(INTEG(70),     IBEG)     ,(INTEG(71),IATSPL(1,1))

EQUIVALENCE (INTEG(13), IOPT)

EQUIVALENCE
    (INTEG(09), INCC),                (INTEG(25), IDREL),
    (TABLE(07), XREL),                (TABLE(08), YREL),
    (TABLE(09), ZREL),                (TABLE(10), PSIREL)

EQUIVALENCE
    (TABLE(11),    TEMPS)     ,(TABLE(13),       SCAL)
   ,(TABLE(20),     FREQ)     ,(TABLE(46),    SQSCALE)
   ,(LOGIC(01), CRTINIT)      ,(LOGIC(10),    PERSPLT)
   ,(LOGIC(11),  PLOTSW)

*****
* * * DIMENSIONS FOR OVERLAY(4,2)
*****

DIMENSION
    AR1X(8),  A1X(8),    AR1Y(8),   A1Y(8),    AR2X(3),   A2X(3),
    AR2Y(3),  A2Y(3),    AR3X(8),   A3X(8),    AR3Y(8),   A3Y(8),
    AR4X(4),  A4X(4),    AR4Y(4),   A4Y(4),    AR5X(10),  A5X(10),
    AR5Y(10), A5Y(10),   AR6X(4),   A6X(4),    AR6Y(4),   A6Y(4),
    AR7X(8),  A7X(8),    AR7Y(8),   A7Y(8),    AR8X(3),   A8X(3),
    AR8Y(3),  A8Y(3),    DR1X(7),   D1X(7),    DR1Y(7),   D1Y(7),
    DR2X(6),  D2X(6),    DR2Y(6),   D2Y(6),    DR3X(8),   D3X(8),
    DR3Y(8),  D3Y(8),    DR4X(10),  D4X(10),   DR4Y(10),  D4Y(10),
    DR5X(8),  D5X(8),    DR5Y(8),   D5Y(8),    DR6X(10),  D6X(10),
    DR6Y(10),      D6Y(10),
    RW9RY(2)       ,     RW15X(2)        ,     RW15Y(2)        ,
    RW21X(2)       ,     RW21Y(2)        ,
    RW9LX(2),      RW9LY(2),       RW9RX(2),
                         TLE(40)         ,     GNX(4)          ,
    GNY(4)         ,     GSX(4)          ,     GSY(4)          ,
    GTNX(4),  GTNY(4),   GTSX(4),   GTSY(4)

*****
* * * DIMENSIONS AND EQUIVALENCES FOR OVERLAY(4,2)
*****

LOGICAL FSS

DIMENSION
    FSS(16)

EQUIVALENCE
    (LDISI(33),     FSS(1))

DIMENSION
    X(15),     Y(15)

*****
* * * SET WHITE LIGHTS TO INDICATE OVERLAY COMBINATIONS IN EXECUTION.
```



```
LDISO(81)=LDISO(83)=.FALSE.

IF(LDISO(82))
   T:  GO TO 10  ->  A2  PG.6
   F:
LDISO(82)=.TRUE.

DO 50  I=1,15
X(I)=0.
50  Y(I)=0.

TLE( 1) = 4RX18
TLE( 2) = 4RXJUL
TLE( 3)=4RX 72
TLE( 4)=4RX
TLE( 5)=4RX
TLE( 6)=4RX TA
TLE( 7)=4RXATM
TLE( 8)=4RX RE
TLE( 9)=4RXAL-
TLE(10)=4RXTIM
TLE(11)=4RXE A
TLE(12)=4RXTC
TLE(13)=4RXSIM
TLE(14)=4RXULA
TLE(15)=4RXTID
TLE(16)=4RXN D
TLE(17)=4RXISP
TLE(18)=4RXLAY
TLE(19)=4RX
TLE(20)=4RX

TLE(21)=4RXDAT
TLE(22)=4RXE:
TLE(23)=4RXRUN
TLE(24)=4RX:
TLE(25)=4RXSCA
TLE(26)=4RXN:
TLE(27)=4RXCON
TLE(28)=4RXTRO
TLE(29)=4RXLLE
TLE(30)=4RXR
TLE(31)=4RX   T
TLE(32)=4RXO A
TLE(33)=4RX/C
```

```
* * * THESE DATA SHOULD BE USED ONLY IN HOLD.I.E. OVERLAY(4,2)
* * * THESE CONSTANTS DEFINE RUNWAYS - 9L,9R,15,AND 21.

RW9LX(1) =  0.34  $  RW9LY(1) = -0.25
RW9LX(2) =  0.38  $  RW9LY(2) =  1.40
RW9RX(1) = -0.38  $  RW9RY(1) = -0.48
RW9RX(2) = -0.33  $  RW9RY(2) =  0.84
RW15X(1) =  0.71  $  RW15Y(1) = -0.14
RW15X(2) = -0.55  $  RW15Y(2) =  0.59
RW21X(1) =  0.65  $  RW21Y(1) =  0.60
RW21X(2) = -0.33  $  RW21Y(2) =  0.03

*****
* * * NORTH AND SOUTH ILS GATE GEOMETRY FOLLOWS

GTNX(1)=-6.10   $ GTNX(2)=-5.98   $ GTNX(3)=-6.98
GTNX(4)=-6.10   $ GTNY(1)=0.0     $ GTNY(2)=0.10
GTNY(3)=0.38    $ GTNY(4)=0.45    $ GTSX(1)=-6.94
GTSX(2)=-5.80   $ GTSX(3)=-5.80   $ GTSX(4)=-5.94
 GTSY(1)=-0.71   $ GTSY(2)=-0.62   $ GTSY(3)=-0.32
GTSY(4)=-0.26

* * * THESE CONSTANTS DEFINE ARRIVAL ROUTES AND DEPARTURE ROUTES

AR1X(1)=0.2      $ AR1X(2)=3.87     $ AR1X(3)=7.9
AR1X(4)=11.3     $ AR1X(5)=15.3     $ AR1X(6)=15.6
AR1X(7)=17.75    $ AR1X(8)=18.9     $ AR1Y(1)=4.4
AR1Y(2)=7.2      $ AR1Y(3)=10.0     $ AR1Y(4)=12.45
AR1Y(5)=18.5     $ AR1Y(6)=20.0     $ AR1Y(7)=30.0
AR1Y(8)=35.15    $ AR2X(1)=15.3     $ AR2X(2)=17.45
AR2X(3)=28.4     $ AR2Y(1)=18.5     $ AR2Y(2)=20.0
AR2Y(3)=28.05    $ AR3X(1)=-0.55    $ AR3X(2)=-0.55
AR3X(3)=-3.28    $ AR3X(4)=-6.0     $ AR3X(5)=-15.65
AR3X(6)=-18.75   $ AR3X(7)=-20.0    $ AR3X(8)=-28.1
AR3Y(1)=5.75     $ AR3Y(2)=8.60     $ AR3Y(3)=9.3
AR3Y(4)=10.0     $ AR3Y(5)=12.4     $ AR3Y(6)=15.0
AR3Y(7)=16.75    $ AR3Y(8)=28.3     $ AR4X(1)=-18.75
AR4X(2)=-20.0    $ AR4X(3)=-30.0    $ AR4X(4)=-37.75
AR4Y(1)=15.0     $ AR4Y(2)=14.9     $ AR4Y(3)=13.85
AR4Y(4)=13.05    $ AR5X(1)=-2.5     $ AR5X(2)=0.15
AR5X(3)=2.8      $ AR5X(4)=5.2      $ AR5X(5)=7.6
AR5X(6)=15.1     $ AR5X(7)=16.5     $ AR5X(8)=20.0
AR5X(9)=30.0     $ AR5X(10)=31.05   $ AR5Y(1)=-3.2
AR5Y(2)=-4.68    $ AR5Y(3)=-6.15    $ AR5Y(4)=-8.08

AR5Y(5)=-10.0    $ AR5Y(6)=-16.05   $ AR5Y(7)=-17.7
AR5Y(8)=-19.5    $ AR5Y(9)=-24.55   $ AR5Y(10)=-25.1
AR6X(1)=16.5     $ AR6X(2)=16.55    $ AR6X(3)=16.9
AR6X(4)=17.15    $ AR6Y(1)=-17.7    $ AR6Y(2)=-20.0
AR6Y(3)=-30.0    $ AR6Y(4)=-36.0    $ AR7X(1)=-2.5
AR7X(2)=-3.3     $ AR7X(3)=-6.6     $ AR7X(4)=-12.5
AR7X(5)=-15.6    $ AR7X(6)=-16.1    $ AR7X(7)=-19.35
AR7X(8)=-20.7    $ AR7Y(1)=-3.2     $ AR7Y(2)=-7.4
AR7Y(3)=-10.0    $ AR7Y(4)=-14.7    $ AR7Y(5)=-18.35
AR7Y(6)=-20.0    $ AR7Y(7)=-30.0    $ AR7Y(8)=-34.1
AR8X(1)=-15.6    $ AR8X(2)=-17.75   $ AR8X(3)=-28.25
AR8Y(1)=-18.35   $ AR8Y(2)=-20.0    $ AR8Y(3)=-28.15

* * * BEGIN THE DEPARTURE ROUTES
```



```
DR1X(1)=4.7      $ DR1X(2)=7.5      $ DR1X(3)=10.0
DR1X(4)=11.5     $ DR1X(5)=20.0     $ DR1X(6)=30.0
DR1X(7)=38.6     $ DR1Y(1)=0.4      $ DR1Y(2)=0.45
DR1Y(3)=0.5      $ DR1Y(4)=0.5      $ DR1Y(5)=3.55
DR1Y(6)=7.15     $ DR1Y(7)=10.3
DR2X(1)=DR2X(2)=DR2X(3)=DR2X(4)=DR2X(5)=DR2X(6)=2.8
DR2Y(1)=-4.0     $ DR2Y(2)=-7.0     $ DR2Y(3)=-10.0
DR2Y(4)=-20.0    $ DR2Y(5)=-30.0    $ DR2Y(6)=-39.8
DR3X(1)=4.7      $ DR3X(2)=7.0      $ DR3X(3)=6.0
DR3X(4)=4.85     $ DR3X(5)=3.0      $ DR3X(6)=1.15
DR3X(7)=-2.5     $ DR3X(8)=-6.0     $ DR3Y(1)=0.4
DR3Y(2)=4.3      $ DR3Y(3)=7.0      $ DR3Y(4)=10.0
DR3Y(5)=15.0     $ DR3Y(6)=20.0     $ DR3Y(7)=30.0
DR3Y(8)=39.4     $ DR4X(1)=7.0      $ DR4X(2)=3.8
DR4X(3)=0.9      $ DR4X(4)=-2.6     $ DR4X(5)=-7.0
DR4X(6)=-10.0    $ DR4X(7)=-15.3    $ DR4X(8)=-20.0
DR4X(9)=-30.     $ DR4X(10)=-39.9   $ DR4Y(1)=4.3
DR4Y(2)=6.1      $ DR4Y(3)=7.8      $ DR4Y(4)=7.8
DR4Y(5)=7.8      $ DR4Y(6)=7.8      $ DR4Y(7)=7.8
DR4Y(8)=5.85     $ DR4Y(9)=1.65     $ DR4Y(10)=-2.6

DR5X(1)=2.8      $ DR5X(2)=6.4      $ DR5X(3)=10.0
DR5X(4)=15.0     $ DR5X(5)=20.0     $ DR5X(6)=27.2
DR5X(7)=30.0     $ DR5X(8)=40.0     $ DR5Y(1)=-4.0
DR5Y(2)=-3.55    $ DR5Y(3)=-3.1     $ DR5Y(4)=-2.5
DR5Y(5)=-1.85    $ DR5Y(6)=-1.0     $ DR5Y(7)=-0.92
DR5Y(8)=-0.75    $ DR6X(1)=2.8      $ DR6X(2)=0.8
DR6X(3)=-1.1     $ DR6X(4)=-5.6     $ DR6X(5)=-10.0
DR6X(6)=-14.3    $ DR6X(7)=-20.0    $ DR6X(8)=-28.7
DR6X(9)=-30.0    $ DR6X(10)=-39.9   $ DR6Y(1)=-4.0
DR6Y(2)=-7.0     $ DR6Y(3)=-10.0    $ DR6Y(4)=-10.0
DR6Y(5)=-10.0    $ DR6Y(6)=-10.0    $ DR6Y(7)=-6.7
DR6Y(8)=-1.7     $ DR6Y(9)=-1.8     $ DR6Y(10)=-2.3
```

A2 PG.4

```
10
CONTINUE

**********************************************************************
C**** HOLD SECTION ****************************************************
**********************************************************************

NFRAMF=32.*FREQ+.5
SCAL=50.0/FLOAT(ISCAL)
C=SCAL

CALL
CALL DSPLAY

IF(.NOT.PLOTSW
.OR..NOT.FSS(1
5))
F
T
GO
TO
98765
```

A3 PG.7


```
CALL
CALL HALT

CALL
CALL MAKTAPE(10)

CALL
CALL UNLODE

PLOTSW=.FALSE.

CALL
CALL READY
```

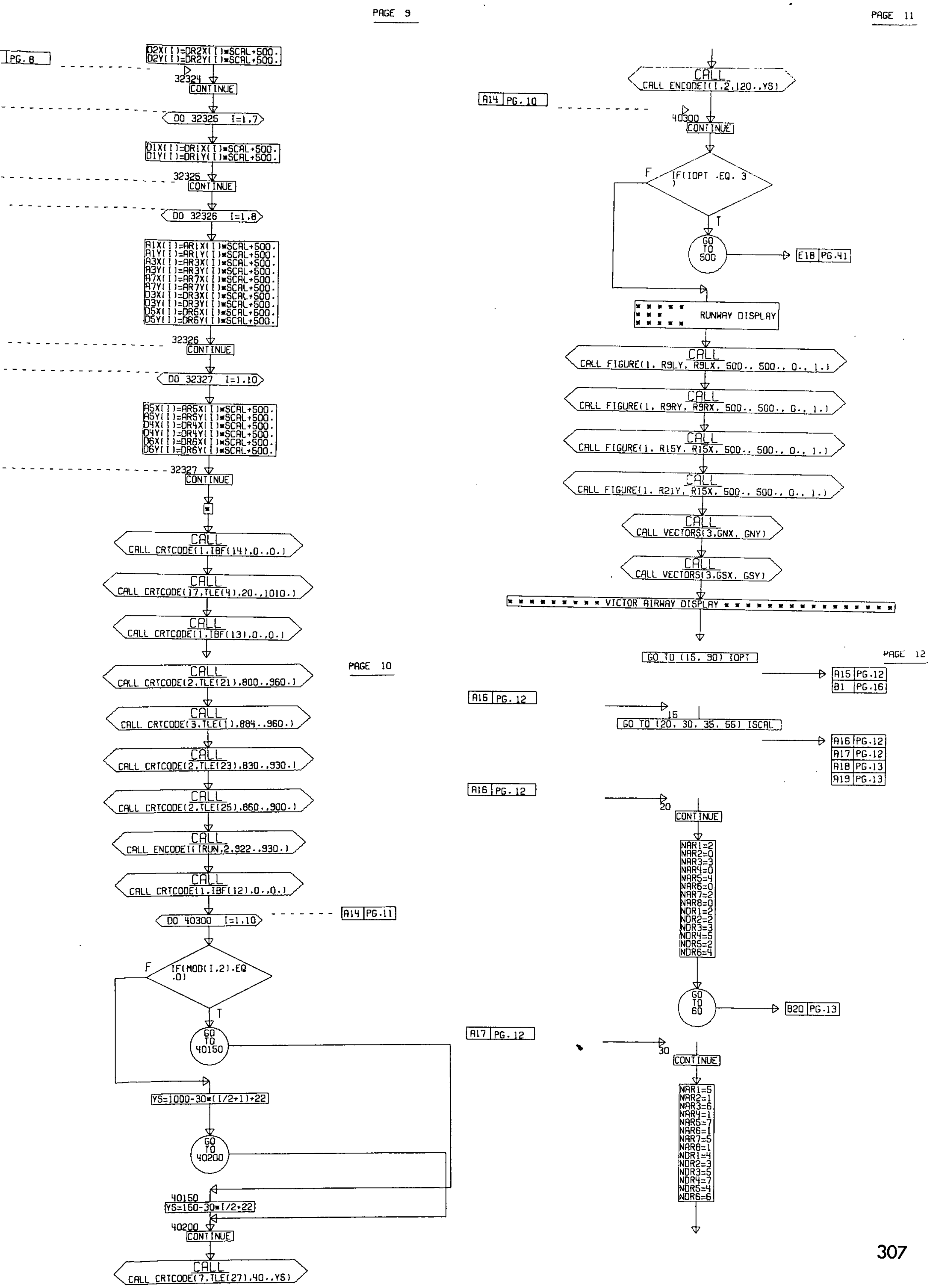
A3 PG.6

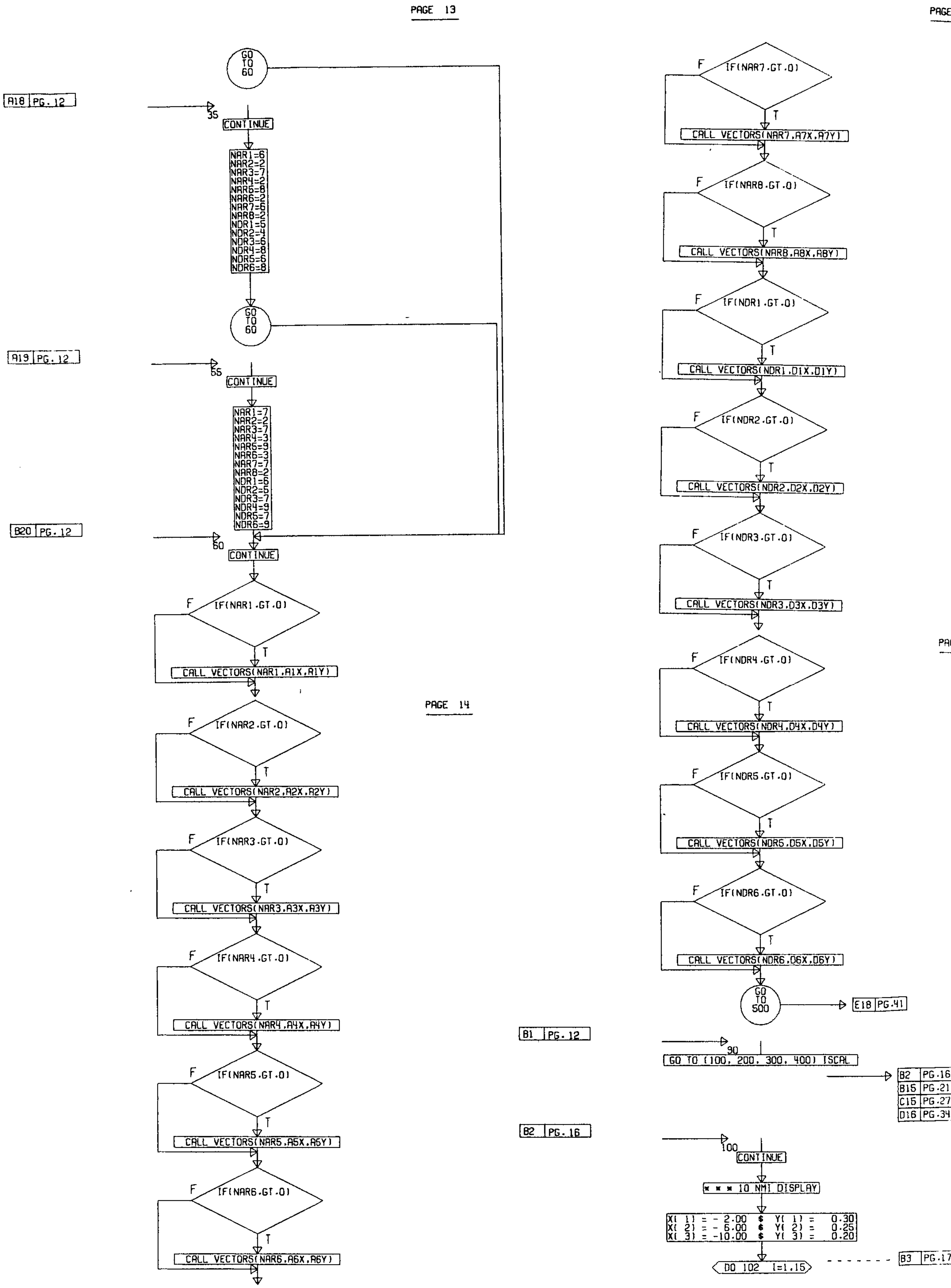
```
98765
CONTINUE

IF(.NOT.FSS(9)
.OR..NOT.CRTIN
IT)
F
T
GO
TO
6
```

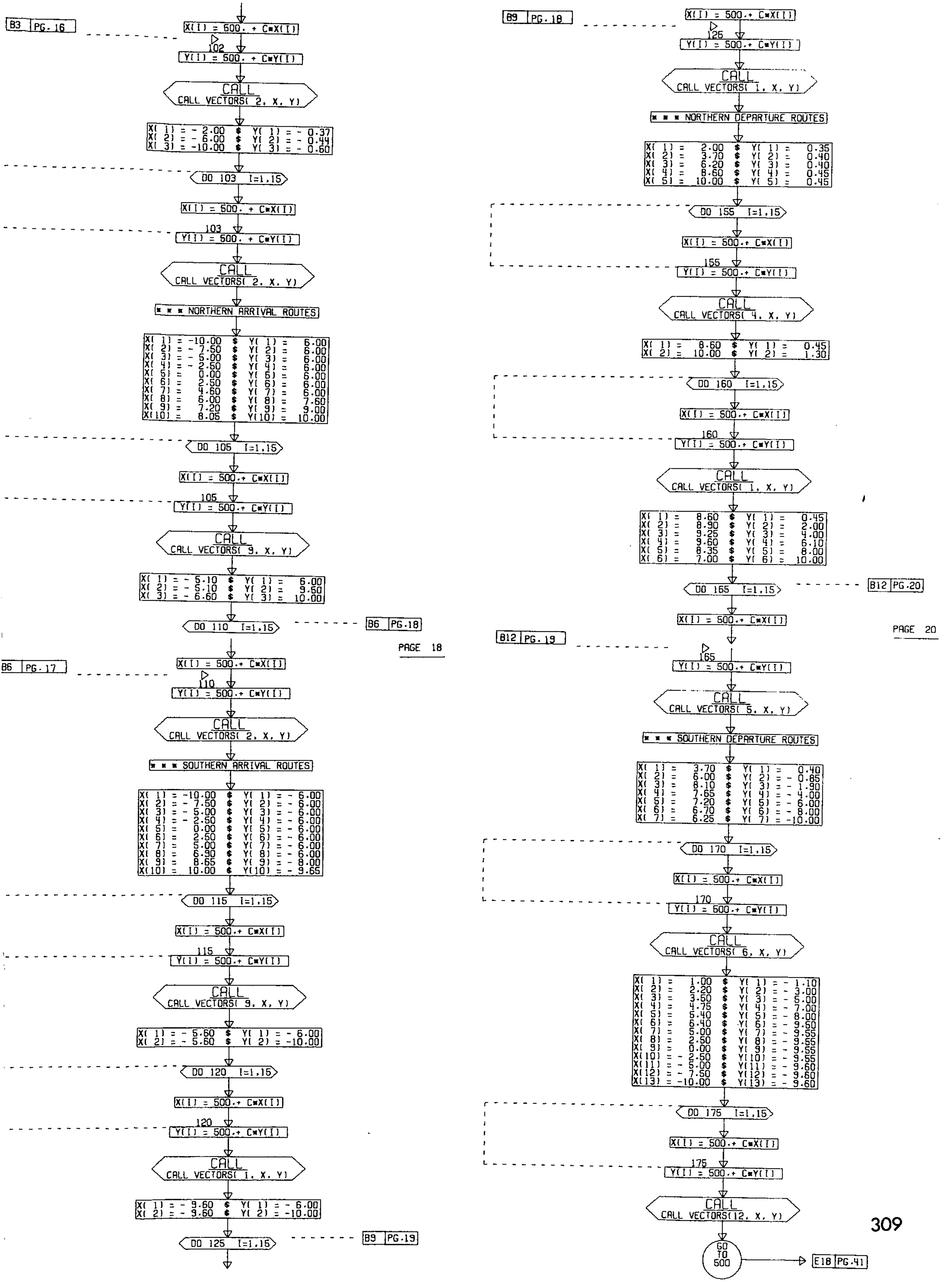
E19 PG.41

```
CALL
CALL HALT

CALL
CALL NAMECRT(6LCRTIPE,ERR)

CALL
CALL READY

40
CALL
CALL SEND

CALL
CALL ENABLE(40S)

CRTINIT=.FALSE.

CALL
CALL HALT

CALL
CALL UNLODE

*
* * * THIS SECTION TRANSLATES THE VICTOR AIRWAY ORIGIN OF COORDINAT
* * * FROM THE CENTER OF THE CRT SCREEN TO THE LOWER LEFT-HAND COR

DO 32321  I=1,2

R9LX(I) = RW9LX(I) * SCAL
R9LY(I) = RW9LY(I) * SCAL
R9RX(I) = RW9RX(I) * SCAL
R9RY(I) = RW9RY(I) * SCAL
R15X(I) = RW15X(I) * SCAL
R15Y(I) = RW15Y(I) * SCAL
R21X(I) = RW21X(I) * SCAL
R21Y(I) = RW21Y(I) * SCAL

32321
CONTINUE

DO 32322  I=1,3

A2X(I)=AR2X(I)*SCAL+500.
A2Y(I)=AR2Y(I)*SCAL+500.
A8X(I)=AR8X(I)*SCAL+500.
A8Y(I)=AR8Y(I)*SCAL+500.

32322
CONTINUE

DO 32323  I=1,4

A4X(I)=AR4X(I)*SCAL+500.
A4Y(I)=AR4Y(I)*SCAL+500.
A6X(I)=AR6X(I)*SCAL+500.
A6Y(I)=AR6Y(I)*SCAL+500.
GNX(I)=GTNX(I)*SCAL+500.
GNY(I)=GTNY(I)*SCAL+500.
GSX(I)=GTSX(I)*SCAL+500.
GSY(I)=GTSY(I)*SCAL+500.

32323
CONTINUE

DO 32324  I=1,6
```

A8 PG


PAGE 9
PG. 8
D2X(I)=DR2X(I)*SCAL+500.
D2Y(I)=DR2Y(I)*SCAL+500.
32324
CONTINUE
DO 32325 I=1,7
D1X(I)=DR1X(I)*SCAL+500.
D1Y(I)=DR1Y(I)*SCAL+500.
32325
CONTINUE
DO 32326 I=1,8
A1X(I)=AR1X(I)*SCAL+500.
A1Y(I)=AR1Y(I)*SCAL+500.
A3X(I)=AR3X(I)*SCAL+500.
A3Y(I)=AR3Y(I)*SCAL+500.
A7X(I)=AR7X(I)*SCAL+500.
A7Y(I)=AR7Y(I)*SCAL+500.
D3X(I)=DR3X(I)*SCAL+500.
D3Y(I)=DR3Y(I)*SCAL+500.
D5X(I)=DR5X(I)*SCAL+500.
D5Y(I)=DR5Y(I)*SCAL+500.
32326
CONTINUE
DO 32327 I=1,10
A5X(I)=AR5X(I)*SCAL+500.
A5Y(I)=AR5Y(I)*SCAL+500.
D4X(I)=DR4X(I)*SCAL+500.
D4Y(I)=DR4Y(I)*SCAL+500.
D6X(I)=DR6X(I)*SCAL+500.
D6Y(I)=DR6Y(I)*SCAL+500.
32327
CONTINUE
CALL
CALL CRTCODE(1,IBF(14),0.,0.)
CALL
CALL CRTCODE(17,TLE(4),20.,1010.)
CALL
CALL CRTCODE(1,IBF(13),0.,0.)
PAGE 10
CALL
CALL CRTCODE(2,TLE(21),800.,960.)
CALL
CALL CRTCODE(3,TLE(1),884.,960.)
CALL
CALL CRTCODE(2,TLE(23),830.,930.)
CALL
CALL CRTCODE(2,TLE(25),860.,900.)
CALL
CALL ENCODEI(IRUN,2,922.,930.)
CALL
CALL CRTCODE(1,IBF(12),0.,0.)
DO 40300 I=1,10
A14 PG.11
F
IF(MOD(I,2).EQ.0)
T
GO TO 40150
YS=1000-30*(I/2+1)+22
GO TO 40200
40150
YS=150-30*I/2+22
40200
CONTINUE
CALL
CALL CRTCODE(7,TLE(27),40.,YS)
PAGE 11
CALL
CALL ENCODEI(I,2,120.,YS)
A14 PG. 10
40300
CONTINUE
F
IF(IOPT .EQ. 3)
T
GO TO 500
E18 PG.41
RUNWAY DISPLAY
CALL
CALL FIGURE(1, R9LY, R9LX, 500., 500., 0., 1.)
CALL
CALL FIGURE(1, R9RY, R9RX, 500., 500., 0., 1.)
CALL
CALL FIGURE(1, R15Y, R15X, 500., 500., 0., 1.)
CALL
CALL FIGURE(1, R21Y, R15X, 500., 500., 0., 1.)
CALL
CALL VECTORS(3,GNX, GNY)
CALL
CALL VECTORS(3,GSX, GSY)
VICTOR AIRWAY DISPLAY
GO TO (15, 90) IOPT
PAGE 12
A15 PG.12
B1 PG.16
A15 PG. 12
15
GO TO (20, 30, 35, 55) ISCAL
A16 PG.12
A17 PG.12
A18 PG.13
A19 PG.13
A16 PG. 12
20
CONTINUE
NAR1=2
NAR2=0
NAR3=3
NAR4=0
NAR5=4
NAR6=0
NAR7=2
NAR8=0
NDR1=2
NDR2=2
NDR3=3
NDR4=5
NDR5=2
NDR6=4
GO TO 60
B20 PG.13
A17 PG. 12
30
CONTINUE
NAR1=5
NAR2=1
NAR3=6
NAR4=1
NAR5=7
NAR6=1
NAR7=5
NAR8=1
NDR1=4
NDR2=3
NDR3=5
NDR4=7
NDR5=4
NDR6=6


PAGE 13
GO TO 60
A18 PG. 12
35
CONTINUE
NAR1=6
NAR2=2
NAR3=7
NAR4=2
NAR5=8
NAR6=2
NAR7=6
NAR8=2
NDR1=5
NDR2=4
NDR3=6
NDR4=8
NDR5=6
NDR6=8
GO TO 60
A19 PG. 12
55
CONTINUE
NAR1=7
NAR2=2
NAR3=7
NAR4=3
NAR5=9
NAR6=3
NAR7=7
NAR8=2
NDR1=6
NDR2=5
NDR3=7
NDR4=9
NDR5=7
NDR6=9
B20 PG. 12
60
CONTINUE
IF(NAR1.GT.0)
F
T
CALL VECTORS(NAR1,A1X,A1Y)
IF(NAR2.GT.0)
CALL VECTORS(NAR2,A2X,A2Y)
IF(NAR3.GT.0)
CALL VECTORS(NAR3,A3X,A3Y)
IF(NAR4.GT.0)
CALL VECTORS(NAR4,A4X,A4Y)
IF(NAR5.GT.0)
CALL VECTORS(NAR5,A5X,A5Y)
IF(NAR6.GT.0)
CALL VECTORS(NAR6,A6X,A6Y)
PAGE 14
IF(NAR7.GT.0)
CALL VECTORS(NAR7,A7X,A7Y)
IF(NAR8.GT.0)
CALL VECTORS(NAR8,A8X,A8Y)
IF(NDR1.GT.0)
CALL VECTORS(NDR1,D1X,D1Y)
IF(NDR2.GT.0)
CALL VECTORS(NDR2,D2X,D2Y)
IF(NDR3.GT.0)
CALL VECTORS(NDR3,D3X,D3Y)
IF(NDR4.GT.0)
CALL VECTORS(NDR4,D4X,D4Y)
IF(NDR5.GT.0)
CALL VECTORS(NDR5,D5X,D5Y)
IF(NDR6.GT.0)
CALL VECTORS(NDR6,D6X,D6Y)
GO TO 500
E18 PG.41
B1 PG. 12
90
GO TO (100, 200, 300, 400) ISCAL
B2 PG.16
B15 PG.21
C15 PG.27
D16 PG.34
B2 PG. 16
100
CONTINUE
* * * 10 NMI DISPLAY
X(1) = - 2.00 $ Y(1) = 0.30
X(2) = - 6.00 $ Y(2) = 0.25
X(3) = -10.00 $ Y(3) = 0.20
DO 102 I=1,15
B3 PG.17


B3 PG.16
X(I) = 500. + C*X(I)
102
Y(I) = 500. + C*Y(I)
CALL
CALL VECTORS(2, X, Y)
X(1) = - 2.00 $ Y(1) = - 0.37
X(2) = - 6.00 $ Y(2) = - 0.44
X(3) = -10.00 $ Y(3) = - 0.60
DO 103 I=1,15
X(I) = 500. + C*X(I)
103
Y(I) = 500. + C*Y(I)
CALL
CALL VECTORS(2, X, Y)
* * * NORTHERN ARRIVAL ROUTES
X(1) = -10.00 $ Y(1) = 6.00
X(2) = - 7.50 $ Y(2) = 6.00
X(3) = - 5.00 $ Y(3) = 6.00
X(4) = - 2.50 $ Y(4) = 6.00
X(5) = 0.00 $ Y(5) = 6.00
X(6) = 2.50 $ Y(6) = 6.00
X(7) = 4.60 $ Y(7) = 6.00
X(8) = 6.00 $ Y(8) = 7.60
X(9) = 7.20 $ Y(9) = 9.00
X(10) = 8.05 $ Y(10) = 10.00
DO 105 I=1,15
X(I) = 500.+ C*X(I)
105
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(9, X, Y)
X(1) = - 5.10 $ Y(1) = 6.00
X(2) = - 5.10 $ Y(2) = 9.50
X(3) = - 6.60 $ Y(3) = 10.00
DO 110 I=1,15
B6 PG.18
PAGE 18
B6 PG.17
X(I) = 500.+ C*X(I)
110
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(2, X, Y)
* * * SOUTHERN ARRIVAL ROUTES
X(1) = -10.00 $ Y(1) = - 6.00
X(2) = - 7.50 $ Y(2) = - 6.00
X(3) = - 5.00 $ Y(3) = - 6.00
X(4) = - 2.50 $ Y(4) = - 6.00
X(5) = 0.00 $ Y(5) = - 6.00
X(6) = 2.50 $ Y(6) = - 6.00
X(7) = 5.00 $ Y(7) = - 6.00
X(8) = 6.90 $ Y(8) = - 6.00
X(9) = 8.55 $ Y(9) = - 8.00
X(10) = 10.00 $ Y(10) = - 9.65
DO 115 I=1,15
X(I) = 500.+ C*X(I)
115
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(9, X, Y)
X(1) = - 5.60 $ Y(1) = - 6.00
X(2) = - 5.60 $ Y(2) = -10.00
DO 120 I=1,15
X(I) = 500.+ C*X(I)
120
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(1, X, Y)
X(1) = - 9.60 $ Y(1) = - 6.00
X(2) = - 9.60 $ Y(2) = -10.00
DO 125 I=1,15
B9 PG.19
B9 PG.18
X(I) = 500.+ C*X(I)
125
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(1, X, Y)
* * * NORTHERN DEPARTURE ROUTES
X(1) = 2.00 $ Y(1) = 0.35
X(2) = 3.70 $ Y(2) = 0.40
X(3) = 6.20 $ Y(3) = 0.40
X(4) = 8.60 $ Y(4) = 0.45
X(5) = 10.00 $ Y(5) = 0.45
DO 155 I=1,15
X(I) = 500.+ C*X(I)
155
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(4, X, Y)
X(1) = 8.60 $ Y(1) = 0.45
X(2) = 10.00 $ Y(2) = 1.30
DO 160 I=1,15
X(I) = 500.+ C*X(I)
160
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(1, X, Y)
X(1) = 8.60 $ Y(1) = 0.45
X(2) = 8.90 $ Y(2) = 2.00
X(3) = 9.25 $ Y(3) = 4.00
X(4) = 9.60 $ Y(4) = 6.10
X(5) = 8.35 $ Y(5) = 8.00
X(6) = 7.00 $ Y(6) = 10.00
DO 165 I=1,15
B12 PG.20
X(I) = 500.+ C*X(I)
PAGE 20
B12 PG.19
165
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(5, X, Y)
* * * SOUTHERN DEPARTURE ROUTES
X(1) = 3.70 $ Y(1) = 0.40
X(2) = 6.00 $ Y(2) = - 0.85
X(3) = 8.10 $ Y(3) = - 1.90
X(4) = 7.65 $ Y(4) = - 4.00
X(5) = 7.20 $ Y(5) = - 6.00
X(6) = 6.70 $ Y(6) = - 8.00
X(7) = 6.25 $ Y(7) = -10.00
DO 170 I=1,15
X(I) = 500.+ C*X(I)
170
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(6, X, Y)
X(1) = 1.00 $ Y(1) = - 1.10
X(2) = 2.20 $ Y(2) = - 3.00
X(3) = 3.50 $ Y(3) = - 5.00
X(4) = 4.75 $ Y(4) = - 7.00
X(5) = 5.40 $ Y(5) = - 8.00
X(6) = 6.40 $ Y(6) = - 9.50
X(7) = 5.00 $ Y(7) = - 9.55
X(8) = 2.50 $ Y(8) = - 9.55
X(9) = 0.00 $ Y(9) = - 9.55
X(10) = - 2.50 $ Y(10) = - 9.55
X(11) = - 5.00 $ Y(11) = - 9.60
X(12) = - 7.50 $ Y(12) = - 9.60
X(13) = -10.00 $ Y(13) = - 9.60
DO 175 I=1,15
X(I) = 500.+ C*X(I)
175
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(12, X, Y)
GO TO 500
E18 PG.41


B15 PG. 16
200 CONTINUE
* * * 20 NMI DISPLAY
X(1) = - 2.00 $ Y(1) = 0.30
X(2) = -11.00 $ Y(2) = 0.15
X(3) = -20.00 $ Y(3) = 0.00
DO 202 I=1,15
X(I) = 500. + C*X(I)
202 Y(I) = 500. + C*Y(I)
CALL
CALL VECTORS(2, X, Y)
X(1) = - 2.00 $ Y(1) = - 0.37
X(2) = -11.00 $ Y(2) = - 0.68
X(3) = -20.00 $ Y(3) = - 0.70
DO 203 I=1,15
X(I) = 500. + C*X(I)
203 Y(I) = 500. + C*Y(I)
CALL
CALL VECTORS(2, X, Y)
* * * NORTHERN ARRIVAL ROUTES
X(1) = -20.00 $ Y(1) = 6.00
X(2) = -15.00 $ Y(2) = 6.00
X(3) = -10.00 $ Y(3) = 6.00
X(4) = - 5.00 $ Y(4) = 6.00
X(5) = 0.00 $ Y(5) = 6.00
X(6) = 4.60 $ Y(6) = 6.00
X(7) = 7.20 $ Y(7) = 9.00
X(8) = 9.75 $ Y(8) = 12.00
X(9) = 12.30 $ Y(9) = 15.00
X(10) = 15.30 $ Y(10) = 18.60
X(11) = 15.60 $ Y(11) = 20.00
DO 205 I=1,15
B18 PG.22
X(I) = 500.+ C*X(I)
B18 PG. 21
205 Y(I) = 500.+ C*Y(I)
PAGE 22
CALL
CALL VECTORS(10, X, Y)
X(1) = 15.30 $ Y(1) = 18.60
X(2) = 17.30 $ Y(2) = 20.00
DO 210 I=1,15
X(I) = 500.+ C*X(I)
210 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(1, X, Y)
X(1) = - 5.10 $ Y(1) = 6.00
X(2) = - 5.10 $ Y(2) = 9.50
X(3) = -10.00 $ Y(3) = 11.20
X(4) = -17.40 $ Y(4) = 13.90
X(5) = -18.70 $ Y(5) = 15.00
X(6) = -20.00 $ Y(6) = 16.80
DO 215 I=1,15
X(I) = 500.+ C*X(I)
215 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(5, X, Y)
X(1) = -12.30 $ Y(1) = 6.00
X(2) = -12.30 $ Y(2) = 9.60
X(3) = -17.40 $ Y(3) = 13.90
DO 220 I=1,15
X(I) = 500.+ C*X(I)
220 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(2, X, Y)
PAGE 23
X(1) = -17.40 $ Y(1) = 6.00
X(2) = -17.40 $ Y(2) = 13.90
X(3) = -18.70 $ Y(3) = 15.00
X(4) = -20.00 $ Y(4) = 14.80
DO 225 I=1,15
X(I) = 500.+ C*X(I)
225 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(3, X, Y)
* * * SOUTHERN ARRIVAL ROUTES
X(1) = -20.00 $ Y(1) = - 6.00
X(2) = -15.00 $ Y(2) = - 6.00
X(3) = -10.00 $ Y(3) = - 6.00
X(4) = - 5.00 $ Y(4) = - 6.00
X(5) = 0.00 $ Y(5) = - 6.00
X(6) = 5.00 $ Y(6) = - 6.00
X(7) = 6.90 $ Y(7) = - 6.00
X(8) = 10.00 $ Y(8) = - 9.60
X(9) = 12.00 $ Y(9) = -12.00
X(10) = 14.00 $ Y(10) = -14.40
X(11) = 16.60 $ Y(11) = -17.40
X(12) = 20.00 $ Y(12) = -19.20
DO 230 I=1,15
X(I) = 500.+ C*X(I)
230 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(11, X, Y)
X(1) = 16.60 $ Y(1) = -17.40
X(2) = 16.70 $ Y(2) = -20.00
DO 235 I=1,15
X(I) = 500.+ C*X(I)
235 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(1, X, Y)
X(1) = - 5.60 $ Y(1) = - 6.00
X(2) = - 5.60 $ Y(2) = -11.40
X(3) = -12.40 $ Y(3) = -15.30
X(4) = -13.70 $ Y(4) = -16.80
X(5) = -14.75 $ Y(5) = -20.00
DO 240 I=1,15
X(I) = 500.+ C*X(I)
240 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(4, X, Y)
X(1) = - 9.60 $ Y(1) = - 6.00
X(2) = - 9.60 $ Y(2) = -12.00
X(3) = -12.40 $ Y(3) = -15.30
DO 245 I=1,15
X(I) = 500.+ C*X(I)
245 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(2, X, Y)
X(1) = -11.90 $ Y(1) = - 6.00
X(2) = -11.90 $ Y(2) = -11.60
X(3) = -12.40 $ Y(3) = -15.30
X(4) = -13.70 $ Y(4) = -16.80
X(5) = -17.80 $ Y(5) = -20.00
DO 250 I=1,15
X(I) = 500.+ C*X(I)
250 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(4, X, Y)
* * * NORTHERN DEPARTURE ROUTES
PAGE

```
X( 1) =   2.00 $ Y( 1) =   0.35
X( 2) =   6.00 $ Y( 2) =   0.40
X( 3) =  10.00 $ Y( 3) =   0.45
X( 4) =  14.00 $ Y( 4) =   0.55
X( 5) =  18.00 $ Y( 5) =   0.60
X( 6) =  20.00 $ Y( 6) =   0.65

DO 265 I=1,15
X(I) = 500.+ C*X(I)
255 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS( 5, X, Y)

X( 1) =   8.60 $ Y( 1) =   0.45
X( 2) =  12.00 $ Y( 2) =   2.40
X( 3) =  15.00 $ Y( 3) =   4.20
X( 4) =  17.60 $ Y( 4) =   5.70
X( 5) =  20.00 $ Y( 5) =   6.50

DO 260 I=1,15
X(I) = 500.+ C*X(I)
260 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS( 4, X, Y)

X( 1) =  17.60 $ Y( 1) =   5.70
X( 2) =  20.00 $ Y( 2) =   5.70

DO 265 I=1,15
X(I) = 500.+ C*X(I)
265 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS( 1, X, Y)
```



```
X( 1) =   8.60 $ Y( 1) =   0.45
X( 2) =   9.10 $ Y( 2) =   3.00
X( 3) =   9.60 $ Y( 3) =   6.10
X( 4) =   7.00 $ Y( 4) =  10.00
X( 5) =   5.00 $ Y( 5) =  13.00
X( 6) =   2.25 $ Y( 6) =  17.00
X( 7) =   0.20 $ Y( 7) =  20.00

DO 270 I=1,15
X(I) = 500.+ C*X(I)
270 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS( 6, X, Y)

* * * SOUTHERN DEPARTURE ROUTES

X( 1) =   3.70 $ Y( 1) =   0.40
X( 2) =   6.00 $ Y( 2) = - 0.85
X( 3) =   8.10 $ Y( 3) = - 1.90
X( 4) =   7.20 $ Y( 4) = - 6.00
X( 5) =   6.40 $ Y( 5) = - 9.50
X( 6) =   5.30 $ Y( 6) = -14.00
X( 7) =   4.10 $ Y( 7) = -19.30
X( 8) =   4.00 $ Y( 8) = -20.00

DO 275 I=1,15
X(I) = 500.+ C*X(I)
275 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS( 7, X, Y)

X( 1) =   4.10 $ Y( 1) = -19.30
X( 2) =   4.60 $ Y( 2) = -20.00

DO 280 I=1,15
X(I) = 500.+ C*X(I)
280 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS( 1, X, Y)
```



```
X( 1) =   1.00 $ Y( 1) = - 1.10
X( 2) =   3.00 $ Y( 2) = - 4.25
X( 3) =   5.40 $ Y( 3) = - 8.00
X( 4) =   6.40 $ Y( 4) = - 9.50
X( 5) =   3.00 $ Y( 5) = - 9.55
X( 6) =   0.00 $ Y( 6) = - 9.55
X( 7) = - 5.00 $ Y( 7) = - 9.55
X( 8) = -10.00 $ Y( 8) = - 9.60
X( 9) = -16.00 $ Y( 9) = - 9.60
X(10) = -20.00 $ Y(10) = - 9.60

DO 285 I=1,15
X(I) = 500.+ C*X(I)
285 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS( 9, X, Y)
GO TO 500  -> E18 PG.41

C15 PG.16 -> 300 CONTINUE
* * * 30 NMI DISPLAY

X( 1) = - 2.00 $ Y( 1) =   0.30
X( 2) = -13.50 $ Y( 2) =   0.15
X( 3) = -25.00 $ Y( 3) =   0.00

DO 302 I=1,15
X(I) = 500. + C*X(I)
302 Y(I) = 500. + C*Y(I)
CALL
CALL VECTORS( 2, X, Y)

X( 1) = - 2.00 $ Y( 1) = - 0.37
X( 2) = -13.50 $ Y( 2) = - 0.66
X( 3) = -25.00 $ Y( 3) = - 0.75

DO 303 I=1,15  - - - C17 PG.28
```

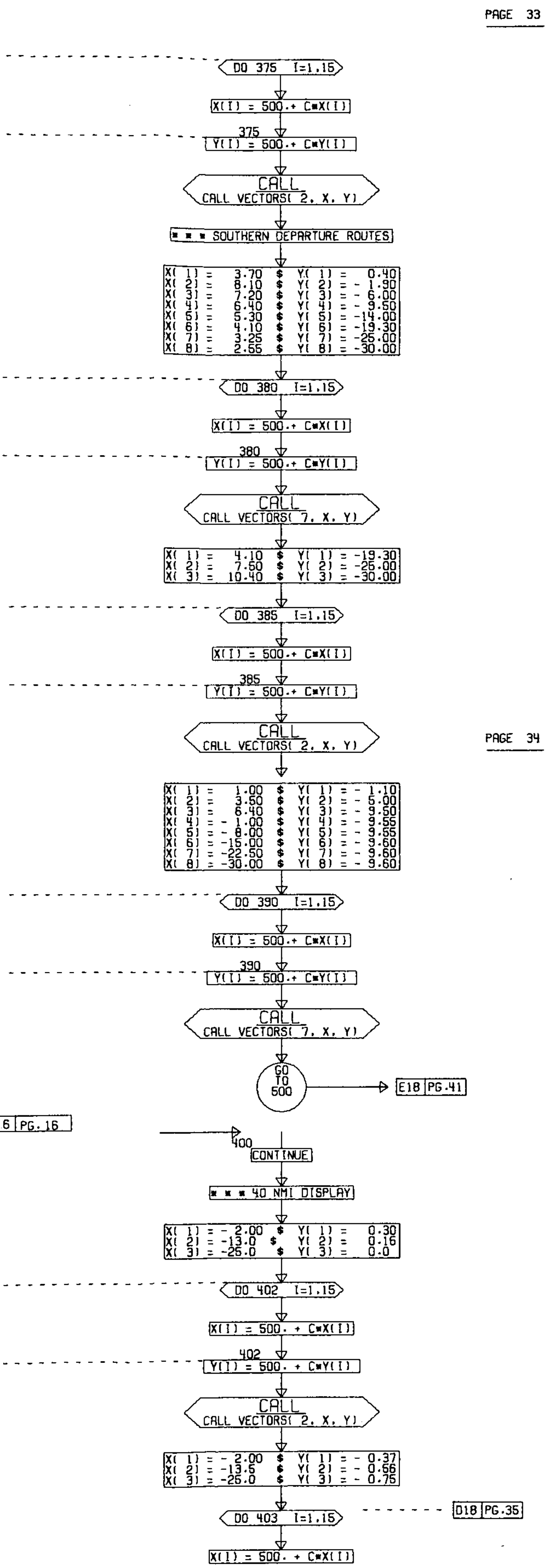
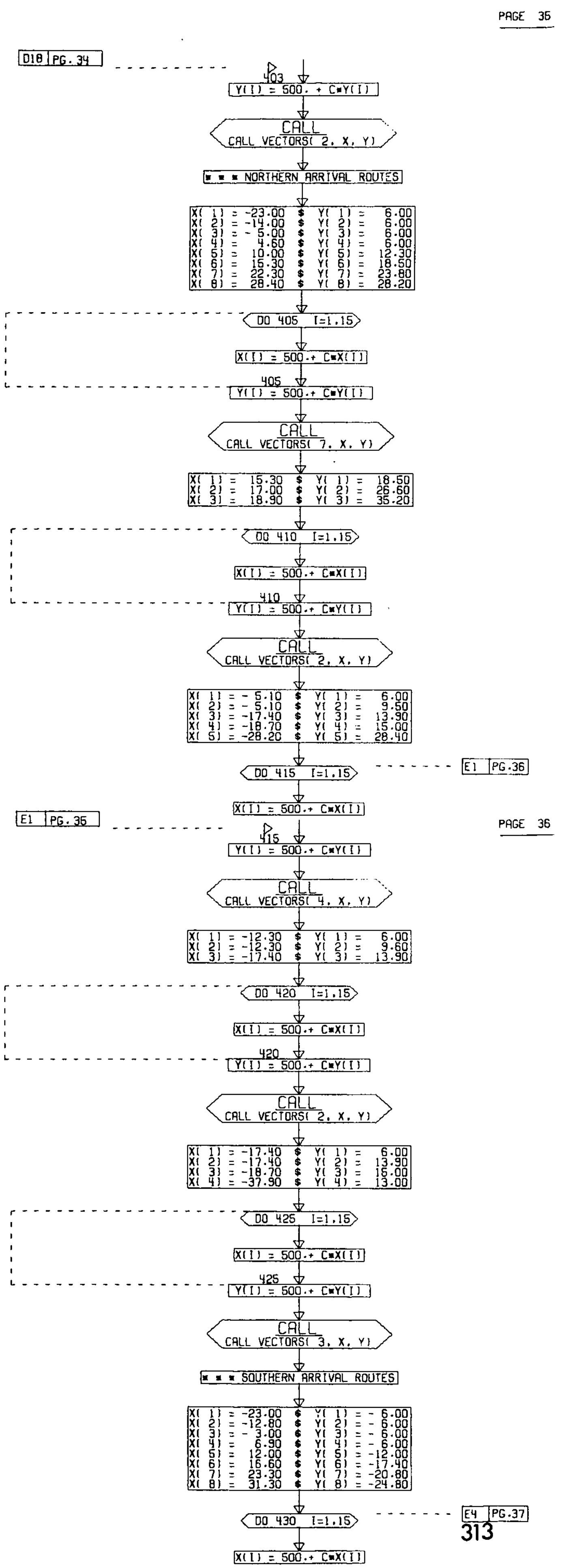


```
C17 PG.27 - - -
X(I) = 500. + C*X(I)
303 Y(I) = 500. + C*Y(I)
CALL
CALL VECTORS( 2, X, Y)

* * * NORTHERN ARRIVAL ROUTES

X( 1) = -23.00 $ Y( 1) =   6.00
X( 2) = -18.40 $ Y( 2) =   6.00
X( 3) = -12.00 $ Y( 3) =   6.00
X( 4) = - 5.00 $ Y( 4) =   6.00
X( 5) =   0.00 $ Y( 5) =   6.00
X( 6) =   4.60 $ Y( 6) =   6.00
X( 7) =   8.90 $ Y( 7) =  11.00
X( 8) =  12.00 $ Y( 8) =  14.65
X( 9) =  15.30 $ Y( 9) =  18.50
X(10) =  16.70 $ Y(10) =  25.00
X(11) =  17.80 $ Y(11) =  30.00

DO 305 I=1,15
X(I) = 500.+ C*X(I)
305 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(10, X, Y)

X( 1) =  15.30 $ Y( 1) =  18.60
X( 2) =  20.00 $ Y( 2) =  22.00
X( 3) =  24.00 $ Y( 3) =  24.90
X( 4) =  28.40 $ Y( 4) =  28.20

DO 310 I=1,15
X(I) = 500.+ C*X(I)
310 Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS( 3, X, Y)
```

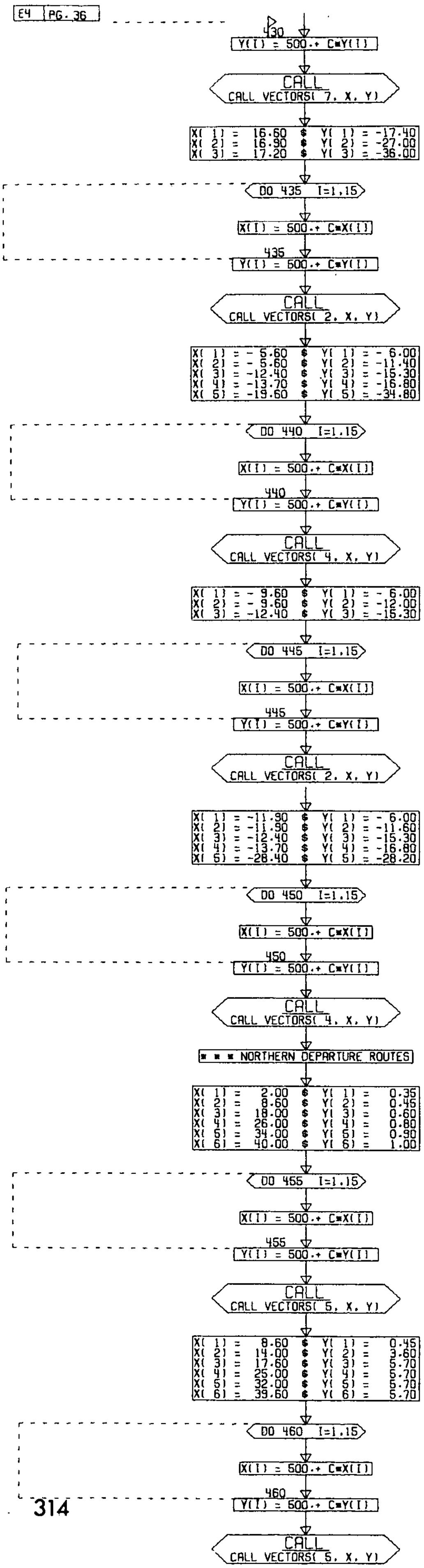
```
X( 1) = - 5.10  $  Y( 1) =   6.00
X( 2) = - 5.10  $  Y( 2) =   9.50
X( 3) = -17.40  $  Y( 3) =  13.90
X( 4) = -18.70  $  Y( 4) =  15.00
X( 5) = -24.00  $  Y( 5) =  22.50
X( 6) = -28.20  $  Y( 6) =  28.40

DO 315  I=1,15
X(I) = 500.+ C*X(I)
315  Y(I) = 500.+ C*Y(I)

CALL
CALL VECTORS( 5, X, Y)

X( 1) = -12.30  $  Y( 1) =   6.00
X( 2) = -12.30  $  Y( 2) =   9.60
X( 3) = -17.40  $  Y( 3) =  13.90

DO 320  I=1,15
X(I) = 500.+ C*X(I)
320  Y(I) = 500.+ C*Y(I)

CALL
CALL VECTORS( 2, X, Y)

X( 1) = -17.40  $  Y( 1) =   6.00
X( 2) = -17.40  $  Y( 2) =  13.90
X( 3) = -18.70  $  Y( 3) =  15.00
X( 4) = -30.00  $  Y( 4) =  13.80

DO 325  I=1,15
X(I) = 500.+ C*X(I)
325  Y(I) = 500.+ C*Y(I)

CALL
CALL VECTORS( 3, X, Y)

* * * SOUTHERN ARRIVAL ROUTES

X( 1) = -23.00  $  Y( 1) = - 6.00
X( 2) = -18.00  $  Y( 2) = - 6.00
X( 3) = -12.80  $  Y( 3) = - 6.00
X( 4) = - 6.40  $  Y( 4) = - 6.00
X( 5) =   0.00  $  Y( 5) = - 6.00
X( 6) =   6.90  $  Y( 6) = - 6.00
X( 7) =  11.80  $  Y( 7) = -11.75
X( 8) =  16.60  $  Y( 8) = -17.40
X( 9) =  23.30  $  Y( 9) = -20.80
X(10) =  30.00  $  Y(10) = -24.10

DO 330  I=1,15
X(I) = 500.+ C*X(I)
330  Y(I) = 500.+ C*Y(I)

CALL
CALL VECTORS( 9, X, Y)

X( 1) =  16.60  $  Y( 1) = -17.40
X( 2) =  16.80  $  Y( 2) = -24.70
X( 3) =  17.00  $  Y( 3) = -30.00

DO 335  I=1,15
X(I) = 500.+ C*X(I)
335  Y(I) = 500.+ C*Y(I)

CALL
CALL VECTORS( 2, X, Y)

X( 1) = - 5.60  $  Y( 1) = - 6.00
X( 2) = - 5.60  $  Y( 2) = -11.40
X( 3) = -12.40  $  Y( 3) = -15.30
X( 4) = -13.70  $  Y( 4) = -16.80
X( 5) = -18.00  $  Y( 5) = -30.00

DO 340  I=1,15
X(I) = 500.+ C*X(I)
340  Y(I) = 500.+ C*Y(I)

CALL
CALL VECTORS( 4, X, Y)
```



```
X( 1) = - 9.60  $  Y( 1) = - 6.00
X( 2) = - 9.60  $  Y( 2) = -12.00
X( 3) = -12.40  $  Y( 3) = -15.30

DO 345  I=1,15
X(I) = 500.+ C*X(I)
345  Y(I) = 500.+ C*Y(I)

CALL
CALL VECTORS( 2, X, Y)

X( 1) = -11.90  $  Y( 1) = - 6.00
X( 2) = -11.90  $  Y( 2) = -11.60
X( 3) = -12.90  $  Y( 3) = -15.30
X( 4) = -13.70  $  Y( 4) = -16.80
X( 5) = -20.00  $  Y( 5) = -21.70
X( 6) = -28.40  $  Y( 6) = -28.20

DO 350  I=1,15
X(I) = 500.+ C*X(I)
350  Y(I) = 500.+ C*Y(I)

CALL
CALL VECTORS( 5, X, Y)

* * * NORTHERN DEPARTURE ROUTES

X( 1) =   2.00  $  Y( 1) =   0.35
X( 2) =   8.60  $  Y( 2) =   0.45
X( 3) =  16.00  $  Y( 3) =   0.60
X( 4) =  23.00  $  Y( 4) =   0.70
X( 5) =  30.00  $  Y( 5) =   0.85

DO 355  I=1,15
X(I) = 500.+ C*X(I)
355  Y(I) = 500.+ C*Y(I)

CALL
CALL VECTORS( 4, X, Y)

X( 1) =   8.60  $  Y( 1) =   0.45
X( 2) =  14.00  $  Y( 2) =   3.60
X( 3) =  17.60  $  Y( 3) =   5.70
X( 4) =  24.00  $  Y( 4) =   5.70
X( 5) =  30.00  $  Y( 5) =   5.70

DO 360  I=1,15
X(I) = 500.+ C*X(I)
360  Y(I) = 500.+ C*Y(I)

CALL
CALL VECTORS( 4, X, Y)

X( 1) =  17.60  $  Y( 1) =   5.70
X( 2) =  24.00  $  Y( 2) =   8.00
X( 3) =  30.00  $  Y( 3) =  10.20

DO 365  I=1,15
X(I) = 500.+ C*X(I)
365  Y(I) = 500.+ C*Y(I)

CALL
CALL VECTORS( 2, X, Y)

X( 1) =   8.60  $  Y( 1) =   0.45
X( 2) =   9.60  $  Y( 2) =   6.10
X( 3) =   5.60  $  Y( 3) =  12.00
X( 4) =   1.50  $  Y( 4) =  18.00
X( 5) = - 3.25  $  Y( 5) =  25.00
X( 6) = - 6.70  $  Y( 6) =  30.00

DO 370  I=1,15
X(I) = 500.+  C*X(I)
370  Y(I) = 500.+  C*Y(I)

CALL
CALL VECTORS( 5, X, Y)

X( 1) =   0.10  $  Y( 1) =  20.20
X( 2) =   0.10  $  Y( 2) =  25.00
X( 3) =   0.10  $  Y( 3) =  30.00
```

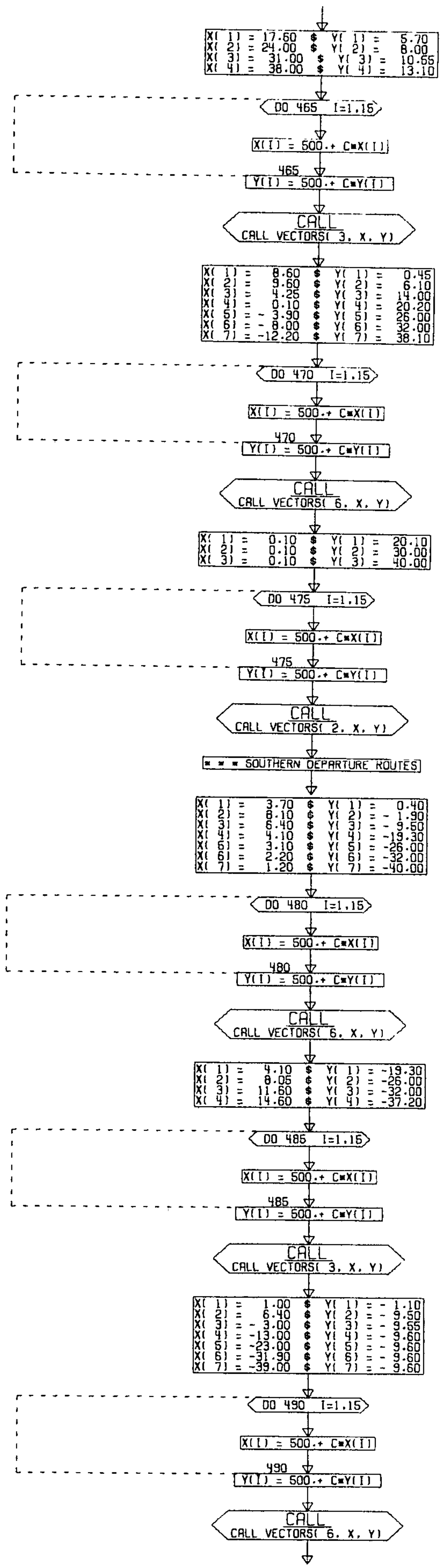

PAGE 33
DO 375 I=1,15
X(I) = 500.+ C*X(I)
375
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(2, X, Y)
* * * SOUTHERN DEPARTURE ROUTES
X(1) = 3.70 $ Y(1) = 0.40
X(2) = 8.10 $ Y(2) = - 1.90
X(3) = 7.20 $ Y(3) = - 6.00
X(4) = 6.40 $ Y(4) = - 9.50
X(5) = 5.30 $ Y(5) = -14.00
X(6) = 4.10 $ Y(6) = -19.30
X(7) = 3.25 $ Y(7) = -25.00
X(8) = 2.55 $ Y(8) = -30.00
DO 380 I=1,15
X(I) = 500.+ C*X(I)
380
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(7, X, Y)
X(1) = 4.10 $ Y(1) = -19.30
X(2) = 7.50 $ Y(2) = -25.00
X(3) = 10.40 $ Y(3) = -30.00
DO 385 I=1,15
X(I) = 500.+ C*X(I)
385
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(2, X, Y)
PAGE 34
X(1) = 1.00 $ Y(1) = - 1.10
X(2) = 3.50 $ Y(2) = - 5.00
X(3) = 6.40 $ Y(3) = - 9.50
X(4) = - 1.00 $ Y(4) = - 9.55
X(5) = - 8.00 $ Y(5) = - 9.55
X(6) = -15.00 $ Y(6) = - 9.60
X(7) = -22.50 $ Y(7) = - 9.60
X(8) = -30.00 $ Y(8) = - 9.60
DO 390 I=1,15
X(I) = 500.+ C*X(I)
390
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(7, X, Y)
GO TO 500
E18 PG.41
6 PG.16
400
CONTINUE
* * * 40 NMI DISPLAY
X(1) = - 2.00 $ Y(1) = 0.30
X(2) = -13.0 $ Y(2) = 0.16
X(3) = -26.0 $ Y(3) = 0.0
DO 402 I=1,15
X(I) = 500. + C*X(I)
402
Y(I) = 500. + C*Y(I)
CALL
CALL VECTORS(2, X, Y)
X(1) = - 2.00 $ Y(1) = - 0.37
X(2) = -13.5 $ Y(2) = - 0.66
X(3) = -26.0 $ Y(3) = - 0.75
DO 403 I=1,15
D18 PG.35
X(I) = 500. + C*X(I)

PAGE 35
D18 PG.34
403
Y(I) = 500. + C*Y(I)
CALL
CALL VECTORS(2, X, Y)
* * * NORTHERN ARRIVAL ROUTES
X(1) = -23.00 $ Y(1) = 6.00
X(2) = -14.00 $ Y(2) = 6.00
X(3) = - 5.00 $ Y(3) = 6.00
X(4) = 4.60 $ Y(4) = 6.00
X(5) = 10.00 $ Y(5) = 12.30
X(6) = 15.30 $ Y(6) = 18.50
X(7) = 22.30 $ Y(7) = 23.80
X(8) = 28.40 $ Y(8) = 28.20
DO 405 I=1,15
X(I) = 500.+ C*X(I)
405
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(7, X, Y)
X(1) = 15.30 $ Y(1) = 18.50
X(2) = 17.00 $ Y(2) = 26.60
X(3) = 18.90 $ Y(3) = 35.20
DO 410 I=1,15
X(I) = 500.+ C*X(I)
410
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(2, X, Y)
X(1) = - 5.10 $ Y(1) = 6.00
X(2) = - 5.10 $ Y(2) = 9.50
X(3) = -17.40 $ Y(3) = 13.90
X(4) = -18.70 $ Y(4) = 15.00
X(5) = -28.20 $ Y(5) = 28.40
DO 415 I=1,15
E1 PG.36
X(I) = 500.+ C*X(I)
PAGE 36
E1 PG.35
415
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(4, X, Y)
X(1) = -12.30 $ Y(1) = 6.00
X(2) = -12.30 $ Y(2) = 9.60
X(3) = -17.40 $ Y(3) = 13.90
DO 420 I=1,15
X(I) = 500.+ C*X(I)
420
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(2, X, Y)
X(1) = -17.40 $ Y(1) = 6.00
X(2) = -17.40 $ Y(2) = 13.90
X(3) = -18.70 $ Y(3) = 15.00
X(4) = -37.90 $ Y(4) = 13.00
DO 425 I=1,15
X(I) = 500.+ C*X(I)
425
Y(I) = 500.+ C*Y(I)
CALL
CALL VECTORS(3, X, Y)
* * * SOUTHERN ARRIVAL ROUTES
X(1) = -23.00 $ Y(1) = - 6.00
X(2) = -12.80 $ Y(2) = - 6.00
X(3) = - 3.00 $ Y(3) = - 6.00
X(4) = 6.90 $ Y(4) = - 6.00
X(5) = 12.00 $ Y(5) = -12.00
X(6) = 16.60 $ Y(6) = -17.40
X(7) = 23.30 $ Y(7) = -20.80
X(8) = 31.30 $ Y(8) = -24.80
DO 430 I=1,15
E4 PG.37
X(I) = 500.+ C*X(I)

PAGE 37



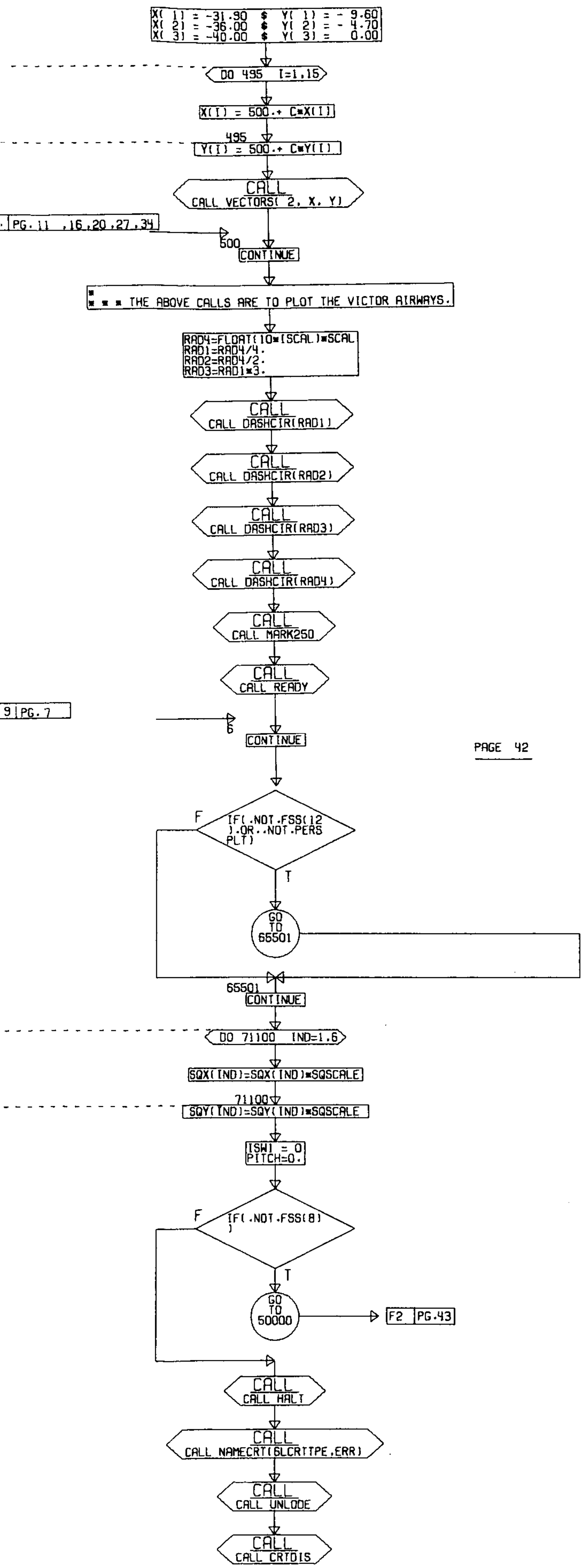
PAGE 38

PAGE

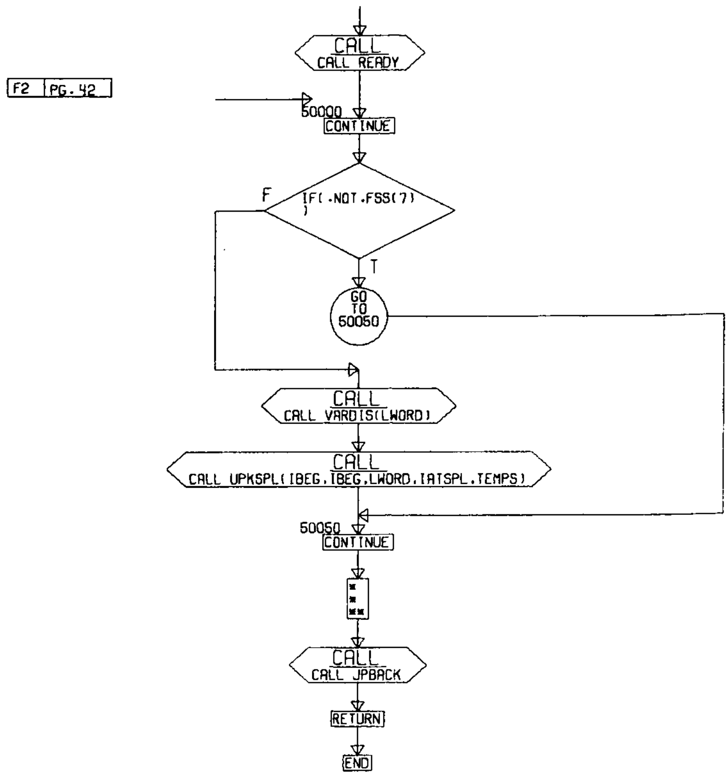


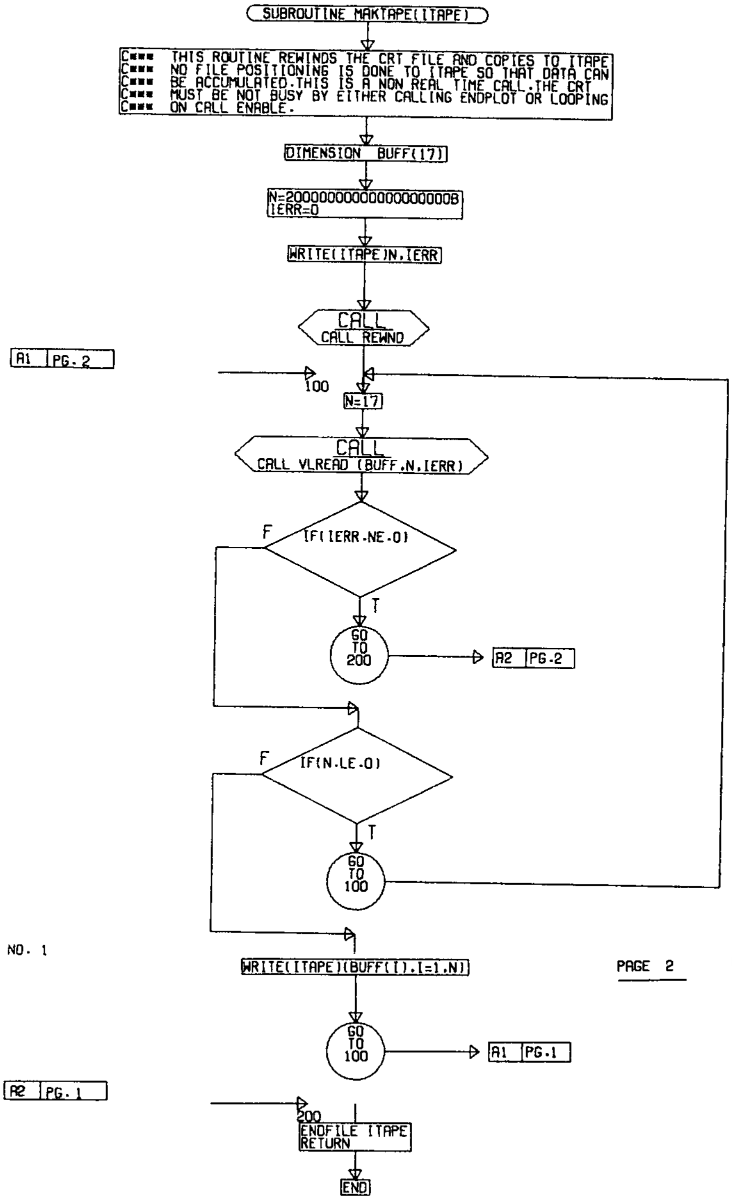
PAGE 4

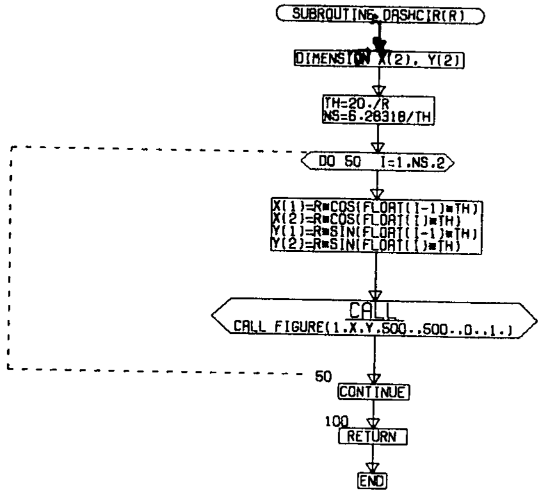
PAGE 41



PAGE 43







CROSSHAIR NO. 1

PAGE 2 IR NO. 1

PAGE 1

```
OVERLAY (TAATMFL.4.3)
PROGRAM TATOPER
```

```
COMMON/OVRBLK/ NTABLE,NINTEG,NLOGIC,NADC,NDAC,NLDISI,NLDISO.
    IJKBUFF(5).      IVARBUF(5).      KTYPE.          VARCHNG
```

```
EQUIVALENCE (INTEG(2).IPL).(INTEG(3).ISL)
LOGICAL LDISI. LDISO
```

```
COMMON / REALTIM/ ADC(32).DAC(64).LDISI(108).LDISO(196).
NOPER.NHOLD.NRESET.NTERM.NPRINT.NREAD
```

```
COMMON/TABLES/
    INTEG(80).       TABLE(075).      LOGIC(20)
```

```
LOGICAL LOGIC
```

```
*****
* * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN
*****
```

```
COMMON/MAJOR1/
    ACLS(50).       ALTOPT(3.7.2). ALTSKD(9).
    ARRMNA.         ARRMXA.        BOUND.
    CGLIDE.         CILS.          CRA20.
    CRA30.          CRB20.         CRB30.
    DGRTD.          DTR.           EHOLDD.
    ENROTA(2).      FTNM.          G.
    GLIDE(2).       HOLDSEP.       IARRN.
    IARRS.          IART(11).      IBF(110).
    ICHKPT(40).     ICNTRA1(85).   ICNTRA2(85).
    IDELT.          IDEPN.         IDEPS.
    IDROUTE(10).    IHPATHE.       IHPATHS.
    INAV(10,5).     IOPTN(55).     IPAPATH(4.5).
    IPASKED(4.7).   IPCHKPT(4.6).  IPCNTR1(4.6).
    IPCNTR2(4.2).   IPDRT(4.5).    IPERFR(3.7).
    IPERFS(4.8).    IPERFT(3).     IPETA1(4.4).
    IPETA2(4.6).    IPFTR(3).      IPMSG2(4.6).
    IPMSG3(4.3).    IPNPATH(4.7).  IPNSITE(4.4).
    IPOPL(2).       IPOPPA.        IPOPR(2).
    IPOPTN(4.9).    IPSKED(4.8).   IPTAS(4.5)
```

```
COMMON/MAJOR2/
    IPTRAF1(4.4).   IPTRAF2(4.9).  ISDST(17).
    ISKDPU.         ISKED(85).     ISTARTA(21).
    ITANG(2.2).    .ITRAF1(102).   ITRAF2(102).
    MAXK.           MF.
    NALTSKD.        NCP.           NDEPR.
    NNAV(5).        NNPATH(50).    NOUT(20).
    NSPDW.          OUT(26).       PI.
    PPMX(3).        PPMY(3).       RCP(2).
    RTD.            RWYX(4.2).     RWYY(4.2).
    RWYZ(2).        SHOLDD.        SPOSTD(9.2).
    SQX(6).         SQY(6).        TRIX(5).
    TRIY(5).        XA(2).         XC(2).
    XDILS(2).       XDIST(2).      XFINAL(2).
    XRPOS(2).       YA(2).         YC(2).
    YRPOS(2).       RAD(3).        SIG(3).        IARG
  . IETCOOR(36).    IPETC(4.2).    IACFPER(3)
```

PAGE 2

NO. 1

```
*****
*****
* * * MISCELLANEOUS PRIMARY-TO-PRIMARY AND PRIMARY-TO-SECONDARY COMMON
*****      (0,0). (1,0). (2,0). (3,0). (4,0). (4.1). (4.2). (4.3)
```

```
COMMON/MISC/
    DPRIM.          IIHEM(21).     ITL.           NFRAMF.
    PITCH
```

```
*****
```

```
COMMON/TOES/ IONE.ITWO.ITHREE.IFOUR.IFIVE.ISIX.ISEVEN.IEIGHT.ININE
.ITEN
```

```
*****
* * * REAL-TIME MAIN COMMON IN OVERLAY(4.0) AND ALL SECONDARY OVERLAYS A
* * *      PRIMARY SUBROUTINES OF (4.0)
*****
```

```
COMMON/RTMAIN1/
   ASKSPD(35.11).ASPACE1(4.9).ASPACE2(4.9).ASPACE3(4.9).
    CETA.           CONFLCT.       DELH(35).
    HDBEG(35).      HDOWN.         IA(2).
    IAC.            IACTN1(35).    IACTN2(35).
 IACTN3(35). IASKED(35.11). IATAS(35.2).ICNGO.
    IATRAF(35.26).  ICA.           ICACT.
    ICLASS.         ICMSG1(8.5).   ICMSG2(8.5).
    ICMSG3(8.5).    ICOMM(10).     ID(2).
    IDELAY.         IENROT(2.20).  IETAS(35.2).
    IETA1(36.10).   IETA2(36.10).  IE360(60).
    IFCMSG1(8.25).  IFCMSG2(8.25). IFCMSG3(8.25).
    IHOWN.          IOBETA.        IOEETA.
    IOPATH.         IPASSAC.       ISEQ.
    ISEQUE(3.16).   ISW4.          ITOQUE(2.20).
    ITRY.           IWHICH.        KNTP.
    L.              LEMG(5).       NASPACE(4).
    NCMSG(10).      NEMG.          NETA(36).
    NFMSG(10).      NHOLDP(35).    NNA(35).
    NSEQ.           OLDR(35)
```

```
COMMON/RTMAIN2/
    OTHAA.          OTHETA.        OTHOA.
    OTHSETA.        RESTOR(35).    ROLL.
    T(35).          TETAE.         TETAL.
    TETAO.          TIME.          V1.
    V2.             XOWN.          XPL(35).
    YOWN.           YPL(35).       ZOWN
  . R9LX(2).        R9LY(2).       R9RX(2).       R9RY(2).
    R15X(2).        R15Y(2).       R21X(2).       R21Y(2).
    XNREF(4).       YNREF(4)
```

```
*****
*****
* * * DIMENSIONS FOR OVERLAY(4.3)
*****
```

```
DIMENSION
    UPERS(4).       VPERS(4)
  . TYBUF(16)
```

PAGE 3

```
DIMENSION ATRAF(35.26)
EQUIVALENCE (ATRAF(1.1).IATRAF(1.1))
```

```
*****
* * * DIM. + TYPE FOR OVERLAY(4.3)
```

```
DIMENSION      IATSPL(4.1)
LOGICAL        CRTINIT . INITRD . PERSPLT . PLOTSW
```

```
* * * TABLE INTEG LOGIC EQIV. FOR OVERLAY(4.3)
```

```
EQUIVALENCE (INTEG(05). N). (INTEG(06). NFMSGD).
    (INTEG(18). ICNTF)
```

```
EQUIVALENCE
    (INTEG(08).       INC)     .(INTEG(09).       INCC)
   .(INTEG(11).        IO)     .(INTEG(12).       IOUT)
   .(INTEG(23).      NTWO)     .(INTEG(27).        KKK)
   .(INTEG(33).      NCOM)     .(INTEG(34).    NCONTRL)
   .(INTEG(36).    NFRAM)     .(INTEG(38).      NTRAK)
   .(INTEG(39).     NUMSC)     .(INTEG(40).         IT)
   .(INTEG(41).     NSCAN)     .(INTEG(45).       ISTK)
   .(INTEG(67).     ISTOP)     .(INTEG(70).       IBEG)
   .(INTEG(71).IATSPL(1.1))   .(TABLE(11).       TEMPS)
   .(TABLE(29).    UPERSC)     .(TABLE(30).     VPERSC)
   .(TABLE(47).    VELCON)     .(TABLE(53).     CKROLL)
   .(TABLE(54).      CKRC)     .(TABLE(55).      CROLL)
   .(TABLE(56).       CRC)     .(TABLE(57).      FDELT)
   .(TABLE(58).      ZCON)     .(TABLE(59).      HDCON)
   .(TABLE(73).      XSTP)     .(TABLE(74).       YSTP)
   .(TABLE(75).      VSTP)     .(LOGIC(01).    CRTINIT)
   .(LOGIC(08).    INITRD)     .(LOGIC(10).    PERSPLT)
   .(LOGIC(11).    PLOTSW)
```

```
*****
* * * DIMENSIONS AND EQUIVALENCES FOR OVERLAY(4.3)
*****
```

```
LOGICAL FSS
```

```
DIMENSION
    FSS(16).        IXILS(10).       TEMPA(10)
```

```
EQUIVALENCE
    (LDISI(33).     FSS(1)).         (INAV(1.5).     IXILS(1))
```

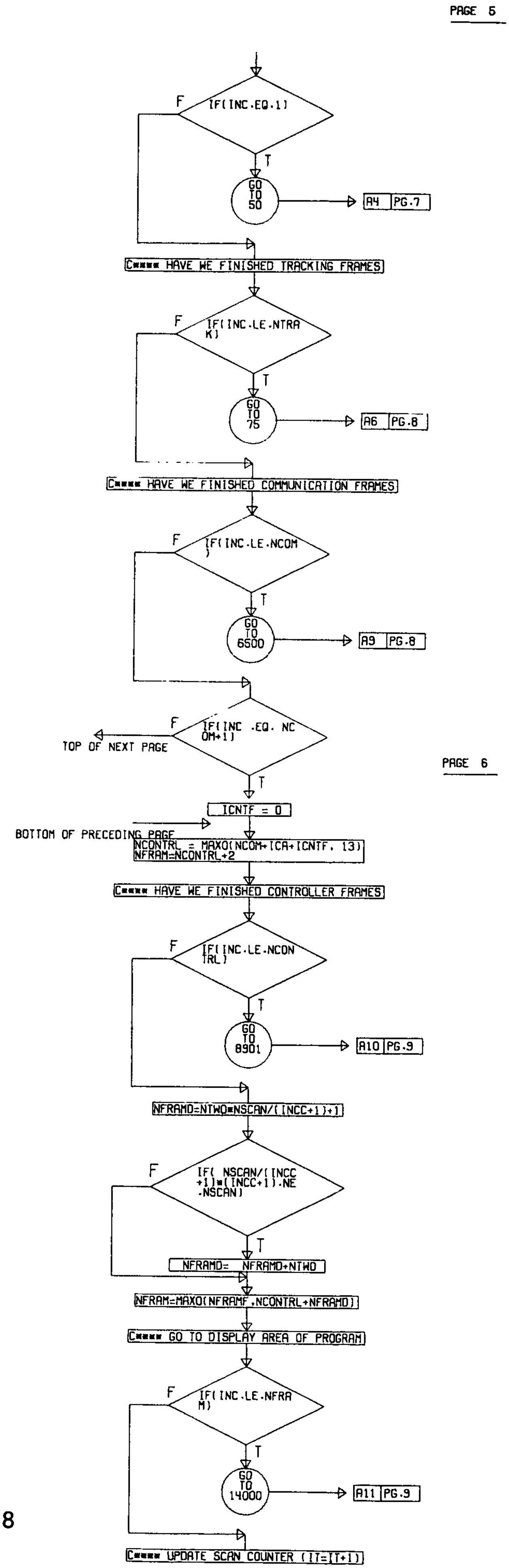
```
*****
* * * SET WHITE LIGHTS TO INDICATE OVERLAY COMBINATIONS IN EXECUTION.
```

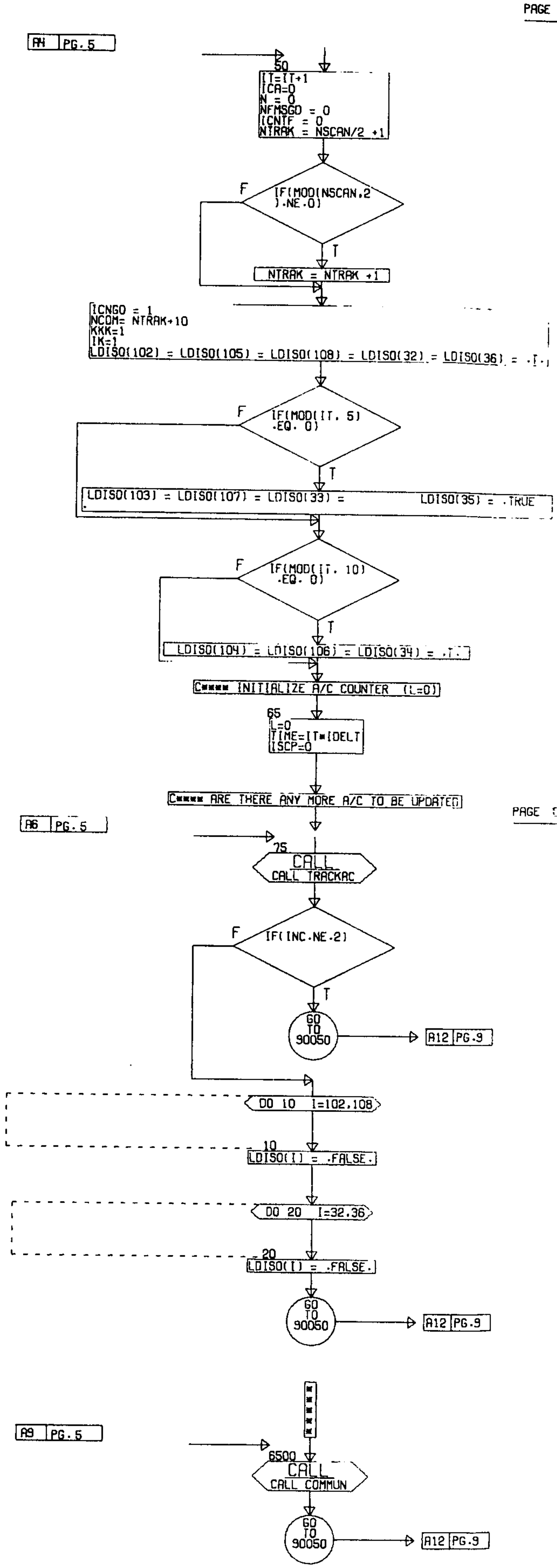
```
LDISO(81)=LDISO(82)=.FALSE.
LDISO(83)=.TRUE.
```

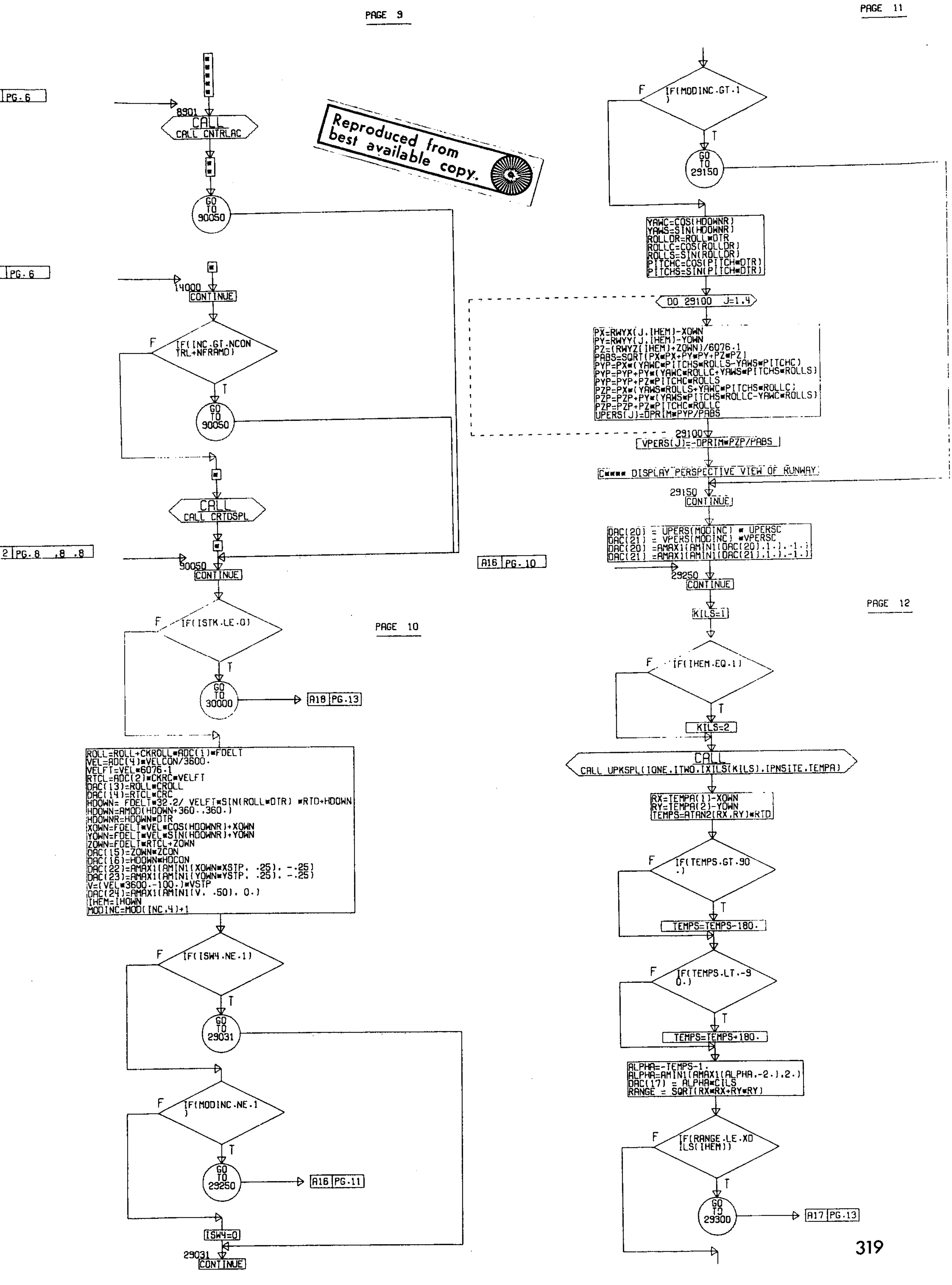
```
**************************************************************
C**** OPERATE LOOP *******************************************
**************************************************************
```

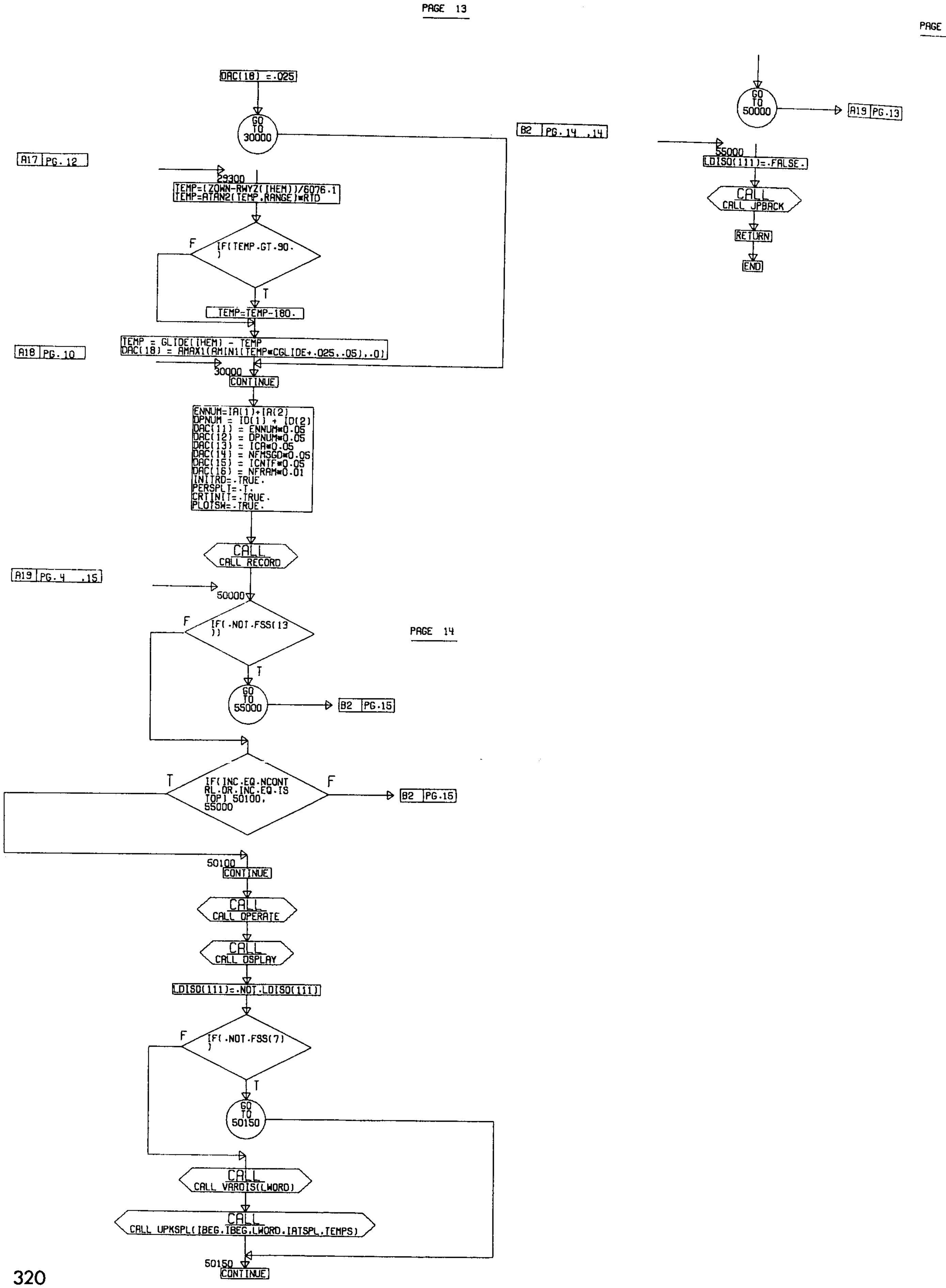
PAGE 4

```
IF(FSS(13))
100, 200                                   T / F

100
IF(INC.EQ.NCONT
RL.OR.INC.EQ.IS
TOP)                                        F / T

GO
TO
50000          A19 PG.13

200 CONTINUE

IOUT=7

DO 50500  IO=1.20

50500
NOUT(IO) = 0
IO=0

IF(ZOWN.LT.RWY
Z(IHOWN))                                   F / T

ISTK=0

INC=MOD(INC.NFRAM)+1

C**** ARE WE BEGINNING A NEW SCAN
```


PAGE 5
IF(INC.EQ.1)
F
T
GO TO 50
A4 PG.7
C**** HAVE WE FINISHED TRACKING FRAMES
IF(INC.LE.NTRAK)
GO TO 75
A6 PG.8
C**** HAVE WE FINISHED COMMUNICATION FRAMES
IF(INC.LE.NCOM)
GO TO 6500
A9 PG.8
IF(INC .EQ. NCOM+1)
TOP OF NEXT PAGE
PAGE 6
ICNTF = 0
BOTTOM OF PRECEDING PAGE
NCONTRL = MAXO(NCOM+ICA+ICNTF, 13)
NFRAM=NCONTRL+2
C**** HAVE WE FINISHED CONTROLLER FRAMES
IF(INC.LE.NCONTRL)
GO TO 8901
A10 PG.9
NFRAMD=NTWO*NSCAN/((NCC+1)+1)
IF(NSCAN/(INCC+1)*(INCC+1).NE.NSCAN)
NFRAMD= NFRAMD+NTWO
NFRAM=MAXO(NFRAMF,NCONTRL+NFRAMD)
C**** GO TO DISPLAY AREA OF PROGRAM
IF(INC.LE.NFRAM)
GO TO 14000
A11 PG.9
C**** UPDATE SCAN COUNTER (IT=IT+1)


A4 PG.5
50
IT=IT+1
ICA=0
N = 0
NFMSGO = 0
ICNTF = 0
NTRAK = NSCAN/2 +1
IF(MOD(NSCAN,2).NE.0)
F
T
NTRAK = NTRAK +1
ICNGO = 1
NCOM= NTRAK+10
KKK=1
IK=1
LDISO(102) = LDISO(105) = LDISO(108) = LDISO(32) = LDISO(36) = .T.
IF(MOD(IT, 5).EQ. 0)
LDISO(103) = LDISO(107) = LDISO(33) = LDISO(35) = .TRUE
IF(MOD(IT, 10).EQ. 0)
LDISO(104) = LDISO(106) = LDISO(34) = .T.
C**** INITIALIZE A/C COUNTER (L=0)
65
L=0
TIME=IT*IDELT
ISCP=0
C**** ARE THERE ANY MORE A/C TO BE UPDATED
A6 PG.5
75
CALL
CALL TRACKAC
IF(INC.NE.2)
GO TO 90050
A12 PG.9
DO 10 I=102,108
10
LDISO(I) = .FALSE.
DO 20 I=32,36
20
LDISO(I) = .FALSE.
GO TO 90050
A12 PG.9
A9 PG.5
6500
CALL
CALL COMMUN
GO TO 90050
A12 PG.9


PAGE 9
PG.6
8901
CALL
CALL CNTRLAC
GO TO 90050
PG.6
14000
CONTINUE
IF(INC.GT.NCON TRL+NFRAMD)
F
T
GO TO 90050
CALL
CALL CRTDSPL
2 PG.8 .8 .8
90050
CONTINUE
IF(ISTK.LE.0)
F
T
GO TO 30000
A18 PG.13
PAGE 10
ROLL=ROLL+CKROLL*ADC(1)*FDELT
VEL=ADC(4)*VELCON/3600.
VELFT=VEL*6076.1
RTCL=ADC(2)*CKRC*VELFT
DAC(13)=ROLL*CROLL
DAC(14)=RTCL*CRC
HDOWN= FDELT*32.2/ VELFT*SIN(ROLL*DTR) *RTD+HDOWN
HDOWN=AMOD(HDOWN+360.,360.)
HDOWNR=HDOWN*DTR
XOWN=FDELT*VEL*COS(HDOWNR)+XOWN
YOWN=FDELT*VEL*SIN(HDOWNR)+YOWN
ZOWN=FDELT*RTCL+ZOWN
DAC(15)=ZOWN*ZCON
DAC(16)=HDOWN*HDCON
DAC(22)=AMAX1(AMIN1(XOWN*XSTP, .25), -.25)
DAC(23)=AMAX1(AMIN1(YOWN*YSTP, .25), -.25)
V=(VEL*3600.-100.)*VSTP
DAC(24)=AMAX1(AMIN1(V, .50), 0.)
IHEM=IHOWN
MODINC=MOD(INC,4)+1
IF(ISW4.NE.1)
F
T
GO TO 29031
IF(MODINC.NE.1)
F
T
GO TO 29250
A16 PG.11
ISW4=0
29031
CONTINUE
A16 PG.10
PAGE 11
IF(MODINC.GT.1)
F
T
GO TO 29150
YAWC=COS(HDOWNR)
YAWS=SIN(HDOWNR)
ROLLDR=ROLL*DTR
ROLLC=COS(ROLLDR)
ROLLS=SIN(ROLLDR)
PITCHC=COS(PITCH*DTR)
PITCHS=SIN(PITCH*DTR)
DO 29100 J=1.4
PX=RWYX(J.IHEM)-XOWN
PY=RWYY(J.IHEM)-YOWN
PZ=(RWYZ(IHEM)+ZOWN)/6076.1
PABS=SQRT(PX*PX+PY*PY+PZ*PZ)
PYP=PX*(YAWC*PITCHS*ROLLS-YAWS*PITCHC)
PYP=PYP+PY*(YAWC*ROLLC+YAWS*PITCHS*ROLLS)
PYP=PYP+PZ*PITCHC*ROLLS
PZP=PX*(YAWS*ROLLS+YAWC*PITCHS*ROLLC)
PZP=PZP+PY*(YAWS*PITCHS*ROLLC-YAWC*ROLLS)
PZP=PZP+PZ*PITCHC*ROLLC
UPERS(J)=DPRIM*PYP/PABS
29100
VPERS(J)=-DPRIM*PZP/PABS
C*** DISPLAY PERSPECTIVE VIEW OF RUNWAY
29150
CONTINUE
DAC(20) = UPERS(MODINC) * UPERSC
DAC(21) = VPERS(MODINC) *VPERSC
DAC(20) =AMAX1(AMIN1(DAC(20).1.).-1.)
DAC(21) =AMAX1(AMIN1(DAC(21).1.).-1.)
29250
CONTINUE
KILS=1
PAGE 12
IF(IHEM.EQ.1)
F
T
KILS=2
CALL
CALL UPKSPL(IONE.ITWO.IXILS(KILS).IPNSITE.TEMPA)
RX=TEMPA(1)-XOWN
RY=TEMPA(2)-YOWN
TEMPS=ATAN2(RX,RY)*RTD
IF(TEMPS.GT.90.)
F
T
TEMPS=TEMPS-180.
IF(TEMPS.LT.-90.)
F
T
TEMPS=TEMPS+180.
ALPHA=-TEMPS-1.
ALPHA=AMIN1(AMAX1(ALPHA,-2.),2.)
DAC(17) = ALPHA*CILS
RANGE = SQRT(RX*RX+RY*RY)
IF(RANGE.LE.XDILS(IHEM))
F
T
GO TO 29300
A17 PG.13


PAGE 13
PAGE
DAC(18) =.025
GO TO 30000
B2 PG.14 .14
GO TO 50000
A19 PG.13
55000
LDISO(111)=.FALSE.
CALL
CALL JPBACK
RETURN
END
A17 PG.12
29300
TEMP=(ZOWN-RWYZ(IHEM))/6076.1
TEMP=ATAN2(TEMP,RANGE)*RTD
F
IF(TEMP.GT.90.
)
T
TEMP=TEMP-180.
TEMP = GLIDE(IHEM) - TEMP
DAC(18) = AMAX1(AMIN1(TEMP*CGLIDE+.025,.05),.0)
A18 PG.10
30000
CONTINUE
ENNUM=IA(1)+IA(2)
DPNUM = ID(1) + ID(2)
DAC(11) = ENNUM*0.05
DAC(12) = DPNUM*0.05
DAC(13) = ICA*0.05
DAC(14) = NFMSGD*0.05
DAC(15) = ICNTF*0.05
DAC(16) = NFRAM*0.01
INITRD=.TRUE.
PERSPLT=.T.
CRTINIT=.TRUE.
PLOTSW=.TRUE.
CALL
CALL RECORD
A19 PG.4 .15
50000
F
IF(.NOT.FSS(13
))
T
PAGE 14
GO TO 55000
B2 PG.15
T
IF(INC.EQ.NCONT
RL.OR.INC.EQ.IS
TOP) 50100,
55000
F
B2 PG.15
50100
CONTINUE
CALL
CALL OPERATE
CALL
CALL DSPLAY
LDISO(111)=.NOT.LDISO(111)
F
IF(.NOT.FSS(7)
)
T
GO TO 50150
CALL
CALL VARDIS(LWORD)
CALL
CALL UPKSPL(IBEG,IBEG,LWORD,IATSPL,TEMPS)
50150
CONTINUE

PAGE 1

```
SUBROUTINE TRACKAC
```

```
LOGICAL LDISI. LDISO
```

```
COMMON / REALTIM/ ADC(32).DAC(64).LDISI(108).LDISO(196).
NOPER.NHOLD.NRESET.NTERM.NPRINT.NREAD
```

```
COMMON/TABLES/
    INTEG(80).          TABLE(075).       LOGIC(20)
```

```
LOGICAL LOGIC
```

```
*****
* * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN
*****
```

```
COMMON/MAJOR1/
    ACLS(50).          ALTOPT(3.7.2).  ALTSKD(9).
    ARRMNA.            ARRMXA.         BOUND.
    CGLIDE.            CILS.           CRA20.
    CRA30.             CRB20.          CRB30.
    DGRTD.             DTR.            EHOLDD.
    ENROTA(2).         FTNM.           G.
    GLIDE(2).          HOLDSEP.        IARRN.
    IARRS.             IART(11).       IBF(110).
    ICHKPT(40).        ICNTRA1(85).    ICNTRA2(85).
    IDELT.             IDEPN.          IDEPS.
    IDROUTE(10).       IHPATHE.        IHPATHS.
    INAV(10.5).        IOPTN(65).      IPAPATH(4.5).
    IPASKED(4.7).      IPCHKPT(4.6).   IPCNTR1(4.6).
    IPCNTR2(4.2).      IPDRT(4.5).     IPERFR(3.7).
    IPERFS(4.8).       IPERFT(3).      IPETA1(4.4).
    IPETA2(4.6).       IPFTR(3).       IPMSG2(4.6).
    IPMSG3(4.3).       IPNPATH(4.7).   IPNSITE(4.4).
    IPOPL(2).          IPOPPA.         IPOPR(2).
    IPOPTN(4.9).       IPSKED(4.8).    IPTAS(4.5)
```

```
COMMON/MAJOR2/
    IPTRAF1(4.4).      IPTRAF2(4.9).   ISDST(17).
    ISKDPU.            ISKED(85).      ISTARTA(21).
    ITANG(2.2).        ITRAF1(102).    ITRAF2(102).
    MAXK.              MF.
    NALTSKD.           NCP.            NDEPR.
    NNAV(5).           NNPATH(50).     NOUT(20).
    NSPDW.             OUT(26).        PI.
    PPMX(3).           PPMY(3).        RCP(2).
    RTD.               RWYX(4.2).      RWYY(4.2).
    RWYZ(2).           SHOLDD.         SPOSTD(9.2).
    SQX(6).            SQY(6).         TRIX(5).
    TRIY(5).           XA(2).          XC(2).
    XDILS(2).          XDIST(2).       XFINAL(2).
    XRPOS(2).          YA(2).          YC(2).
    YRPOS(2).          RAD(3).         SIG(3).         IARG
  . IETCOOR(36).       IPETC(4.2).     IACFPER(3)
```

```
*****
```

PAGE 2

```
COMMON/TOES/ IONE.ITWO.ITHREE.IFOUR.IFIVE.ISIX.ISEVEN.IEIGHT.ININE
.ITEN
```

```
*****
* * * REAL-TIME MAIN COMMON IN OVERLAY(4.0) AND ALL SECONDARY OVERLAYS A
* * *     PRIMARY SUBROUTINES OF (4.0)
*****
```

```
COMMON/RTMAIN1/
   ASKSPD(35.11).ASPACE1(4.9).ASPACE2(4.9).ASPACE3(4.9).
    CETA.              CONFLCT.        DELH(35).
    HDBEG(35).         HDOWN.          IA(2).
    IAC.               IACTN1(35).     IACTN2(35).
 IACTN3(35).  IASKED(35.11).  IATAS(35.2).ICNGO.
    IATRAF(35.26).     ICA.            ICACT.
    ICLASS.            ICMSG1(8.5).    ICMSG2(8.5).
    ICMSG3(8.5).       ICOMM(10).      ID(2).
    IDELAY.            IENROT(2.20).   IETAS(35.2).
    IETA1(36.10).      IETA2(36.10).   IE360(60).
    IFCMSG1(8.25).     IFCMSG2(8.25).  IFCMSG3(8.25).
    IHOWN.             IOBETA.         IOEETA.
    IOPATH.            IPASSAC.        ISEQ.
    ISEQUE(3.16).      ISW4.           ITOQUE(2.20).
    ITRY.              IWHICH.         KNTP.
    L.                 LEMG(5).        NASPACE(4).
    NCMSG(10).         NEMG.           NETA(36).
    NFMSG(10).         NHOLDP(35).     NNA(35).
    NSEQ.              OLDR(35)
```

```
COMMON/RTMAIN2/
    OTHAA.             OTHETA.         OTHOA.
    OTHSETA.           RESTOR(35).     ROLL.
    T(35).             TETAE.          TETAL.
    TETAO.             TIME.           V1.
    V2.                XOWN.           XPL(35).
    YOWN.              YPL(35).        ZOWN
  . R9LX(2).           R9LY(2).        R9RX(2).        R9RY(2).
    R15X(2).           R15Y(2).        R21X(2).        R21Y(2).
    XNREF(4).          YNREF(4)
```

```
*****
* * * DIM. + TYPE FOR TRACKAC
```

```
DIMENSION
 MSG(20).POPUP(13)
  . TYBUF(16)
```

```
* * * TABLE INTEG LOGIC EQIV. FOR  TRACKAC
```

```
EQUIVALENCE (INTEG(05). N)
```

```
EQUIVALENCE
    (INTEG(13). IOPT).                  (INTEG(25). IDREL).
    (TABLE(07). XREL).                  (TABLE(08). YREL).
    (TABLE(09). ZREL).                  (TABLE(10). PSIREL)
```

PAGE 3

```
EQUIVALENCE
    (INTEG(08).      INC)    .(INTEG(12).       IOUT)
   .(INTEG(17).       IR)    .(INTEG(20).       ISW1)
   .(INTEG(21).     ISW2)    .(INTEG(22).       ISW3)
   .(INTEG(40).       IT)    .(INTEG(41).      NSCAN)
   .(INTEG(45).     ISTK)    .(INTEG(46).     MSG(1))
   .(INTEG(66).   IDELET)    .(INTEG(70).       IBEG)
   .(TABLE(01).       AK)    .(TABLE(02).      ALPHK)
   .(TABLE(11).    TEMPS)    .(TABLE(12).     VECERR)
   .(TABLE(14).   DELALT)    .(TABLE(18).     DELSPD)
   .(TABLE(16).     DELT)    .(TABLE(19).      DKCON)
   .(TABLE(25).     PHIK)    .(TABLE(37).       RCON)
   .(TABLE(40).     SCLA)    .(TABLE(41).       SCLE)
   .(TABLE(42).     SCLN)    .(TABLE(46).      RVECT)
   .(TABLE(48).    VECSD)    .(TABLE(60).   POPUP(1))
   .(TABLE(39). RADCP)
```

```
*****
* * * DIMENSION AND EQUIVALENCES FOR TRACKAC
*****
```

```
DIMENSION   IPERFR1(3)
EQUIVALENCE   (IPERFR(1.1). IPERFR1(1))
DIMENSION ITEMPA(10)
```

```
DIMENSION             TEMPA(10).
    ATRAF(35.26).     IATRAF1(35).     ATRAF2(35).      ATRAF3(35).
    ATRAF4(35).       ATRAF5(35).      ATRAF6(35).      IATRAF7(35).
    IATRAF8(35).      IATRAF9(35).     IATRF10(35).     IATRF11(35).
    IATRF12(35).      IATRF16(35).     IATRF17(35).      ATRF18(35).
    ATRF19(35).       ATRF20(35).
    CPCOORD(2).                        FSS(16).         IVOR(10).
    IADF(10).         IFIX(10).        IDME(10).        IXILS(10).
    IPERFR2(3).       IPERFR3(3).      IPERFR4(3).      IPERFR5(3).
    IPERFR6(3).       IPERFR7(3).
    IPERFS1(4).       IPERFS2(4).      IPERFS3(4).      IPERFS6(4).
    IPOPUP(13).       SITE(3).         TMSGA(6).        TSKED(7)
```

```
EQUIVALENCE (ITEMPA.TEMPA)
```

```
EQUIVALENCE
    (IATRAF(1.1).     IATRAF1(1)).     (ATRAF(1.2).      ATRAF2(1)).
    (ATRAF(1.3).      ATRAF3(1)).      (ATRAF(1.4).      ATRAF4(1)).
    (ATRAF(1.5).      ATRAF5(1)).      (ATRAF(1.6).      ATRAF6(1)).
    (IATRAF(1.7).     IATRAF7(1)).     (IATRAF(1.8).     IATRAF8(1)).
    (IATRAF(1.9).     IATRAF9(1)).     (IATRAF(1.10).    IATRF10(1)).
    (IATRAF(1.11).    IATRF11(1)).     (IATRAF(1.12).    IATRF12(1)).
    (IATRAF(1.16).    IATRF16(1)).     (IATRAF(1.17).    IATRF17(1)).
    (IATRAF(1.18).     ATRF18(1)).     (IATRAF(1.1).     ATRAF(1.1)).
    (IATRAF(1.19).    ATRF19(1)).      (IATRAF(1.20).    ATRF20(1)).
    (CPCOORD(1).      XCP).            (CPCOORD(2).      YCP).
    (LDISI(33).       FSS(1))
```

```
EQUIVALENCE
    (NNAV(3).         NFIX).           (INAV(1.1).       IVOR(1)).
    (INAV(1.2).       IADF(1)).        (INAV(1.3).       IFIX(1)).
    (INAV(1.4).       IDME(1)).        (INAV(1.5).       IXILS(1)).
    (IPERFR(1.2).     IPERFR2(1)).     (IPERFR(1.3).     IPERFR3(1)).
    (IPERFR(1.4).     IPERFR4(1)).     (IPERFR(1.5).     IPERFR5(1)).
    (IPERFR(1.6).     IPERFR6(1)).     (IPERFR(1.7).     IPERFR7(1)).
    (IPERFS(1.1).     IPERFS1(1)).     (IPERFS(1.2).     IPERFS2(1)).
    (IPERFS(1.3).     IPERFS3(1)).     (IPERFS(1.6).     IPERFS6(1)).
    (IPOPUP(1).       POPUP(1))
```

```
* * * CONTINUATION OF EQUIVALENCES FOR TRACKAC
```

PAGE 4

```
EQUIVALENCE
    (SITE(1).         CPX).            (SITE(2).         CPY).
    (SITE(3).         S).              (TMSGA(1).        XDENT).
    (TMSGA(2).        XMSGL).          (TMSGA(3).        XLTT).
    (TMSGA(4).        XCLEAR).         (TSKED(1).        XMODE).
    (TSKED(2).        XINFO1).         (TSKED(3).        XINFO2).
    (TSKED(4).        XFALT).          (TSKED(5).        XMODECP).
    (TSKED(6).        XINFO3).         (TSKED(7).        XINFO4)
```

```
LOGICAL FSS
```

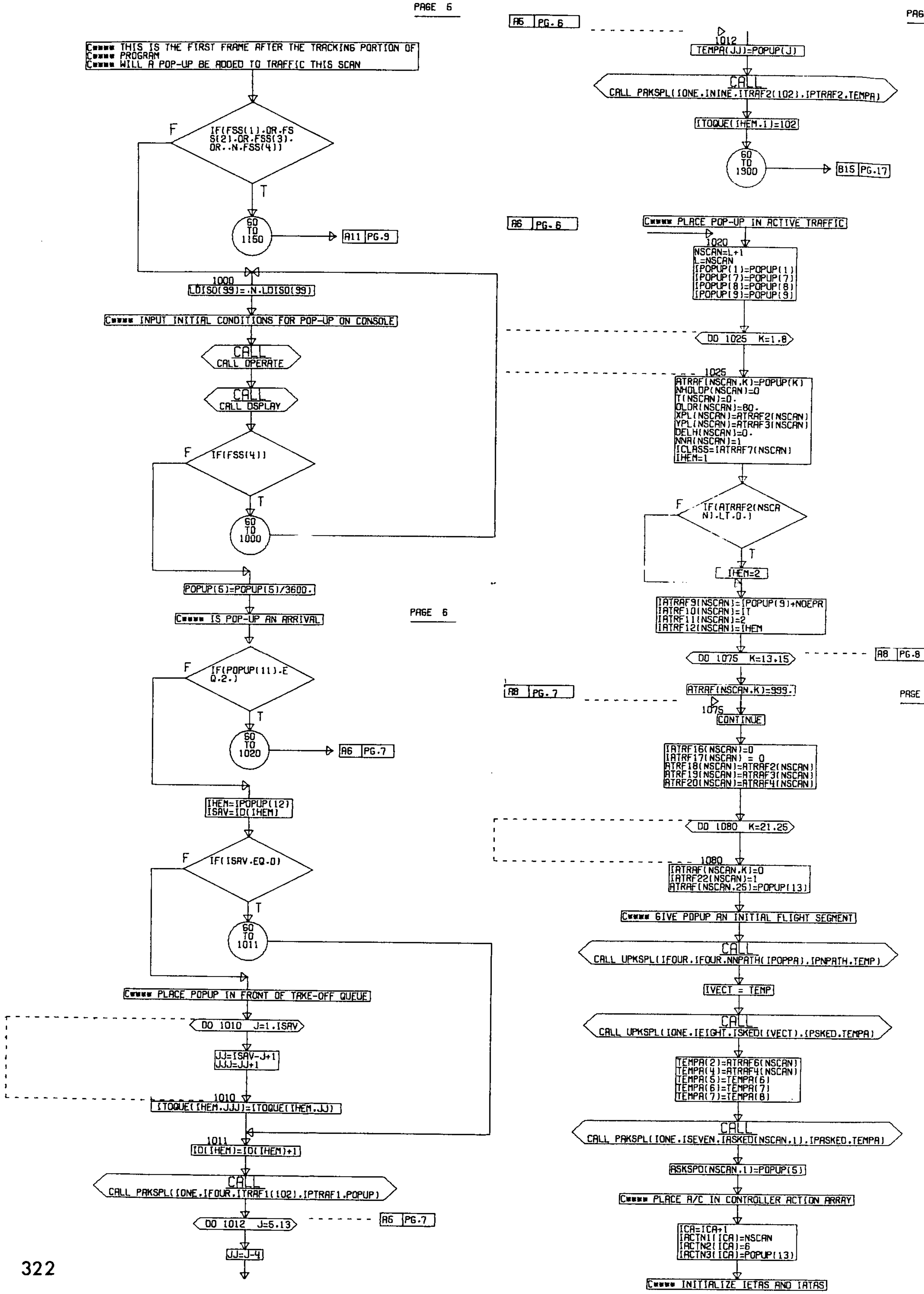
```
DIMENSION
    IATRF21(35).      IATRF22(35).     IATRF23(35).     IATRF24(35).
    IATRF25(35).      IOPTION(9).      OPTION(9)
```

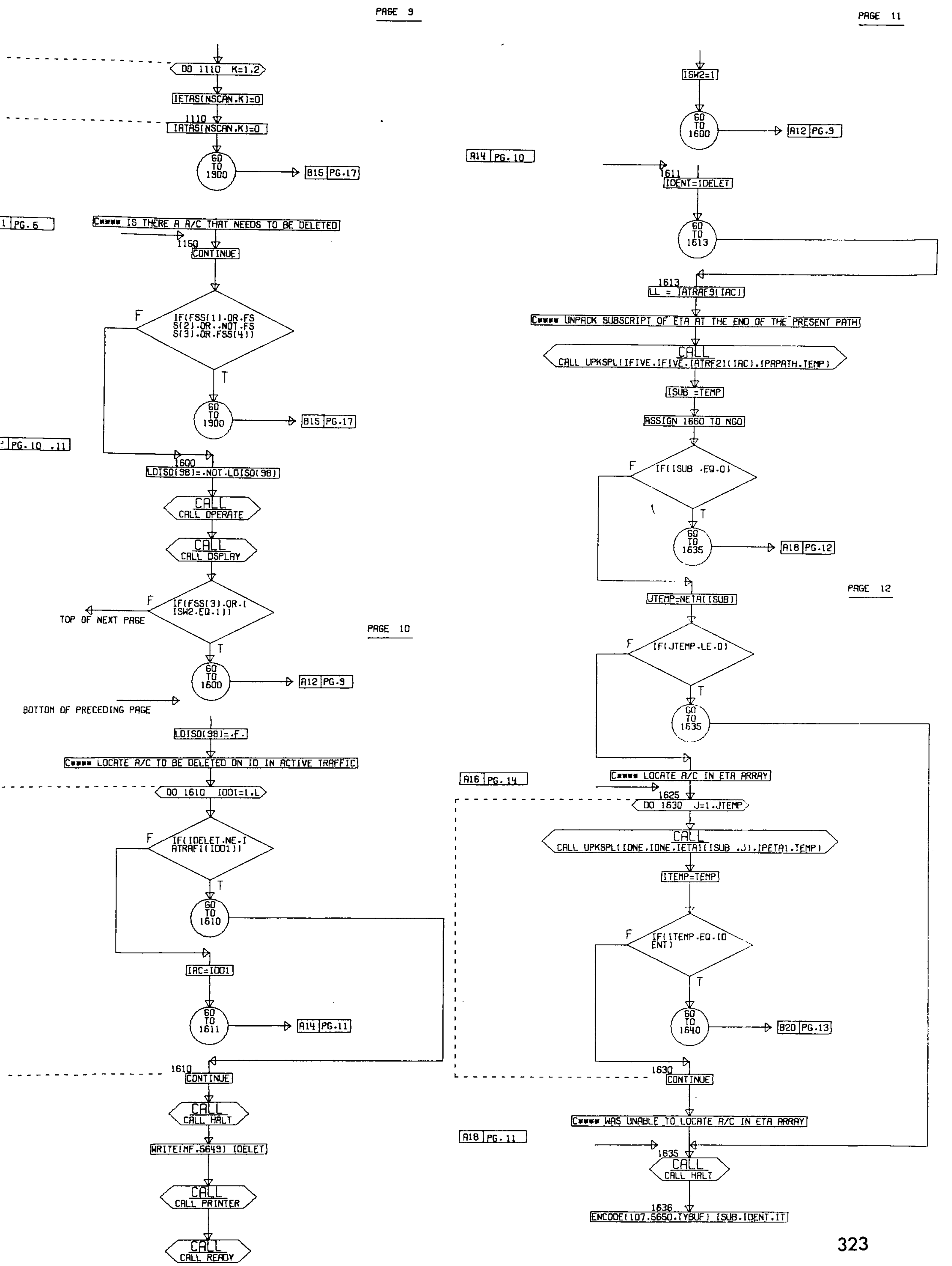
```
EQUIVALENCE
    (IATRAF(1.21). IATRF21(1)).        (IATRAF(1.22). IATRF22(1)).
    (IATRAF(1.23). IATRF23(1)).        (IATRAF(1.24). IATRF24(1)).
    (IATRAF(1.25). IATRF25(1))
```

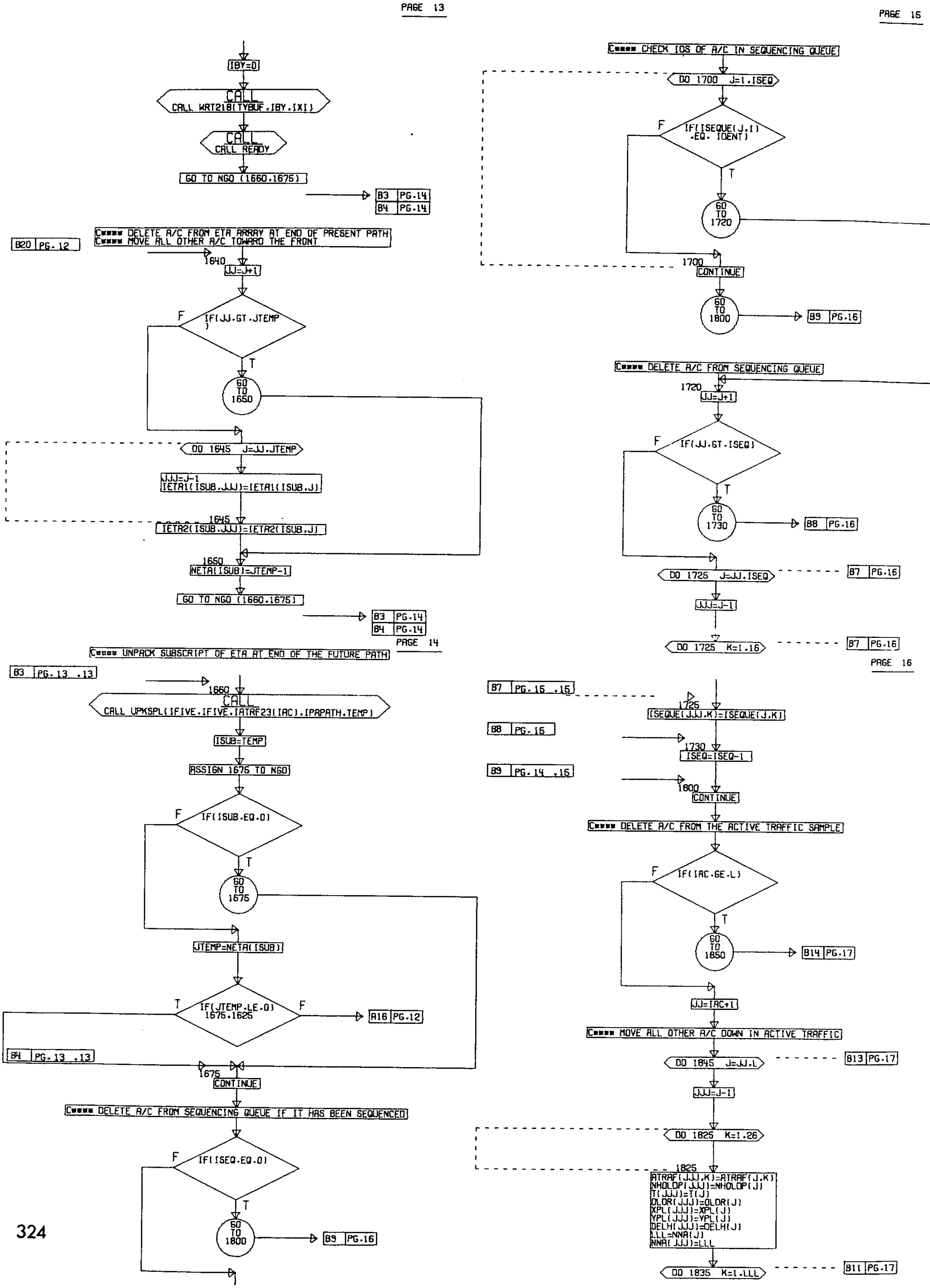
```
EQUIVALENCE
    (RESTOR(01). TETA).                (RESTOR(02). NUMPATH).
    (RESTOR(03). OALT).                (RESTOR(04). AALT).
    (RESTOR(05). ETASEP).              (RESTOR(06). FVEL).
    (RESTOR(07). DSPD).                (RESTOR(08). TX).
    (RESTOR(09). TOTD).                (RESTOR(10). NSEGM).
    (RESTOR(11). DST1).                (RESTOR(12). DST2).
    (RESTOR(13). DST3).                (RESTOR(14). ISUBCK).
    (RESTOR(15). ISUBOP).              (RESTOR(16). ICACT2).
    (RESTOR(17). STEMP).               (RESTOR(18). ISUBN).
    (RESTOR(19). ISUBO).               (RESTOR(20). SEP).
    (RESTOR(21). XLA).                 (RESTOR(22). XLB).
    (RESTOR(23). IASPCON).
    (RESTOR(24). IOALT. IOPTION(1). OPTION(1)).
    (RESTOR(25). IOSPD).               (RESTOR(26). IVECT).
    (RESTOR(27). IVECOP).              (RESTOR(28). IOVERG2).
    (RESTOR(29). IALTOP).              (RESTOR(30). ISPDOP).
    (RESTOR(31). ISHOP).               (RESTOR(32). IEHOP)
  . (RESTOR(33). OSPD)
  . (RESTOR(34). TX3).                 (RESTOR(35). INDSPD)
```

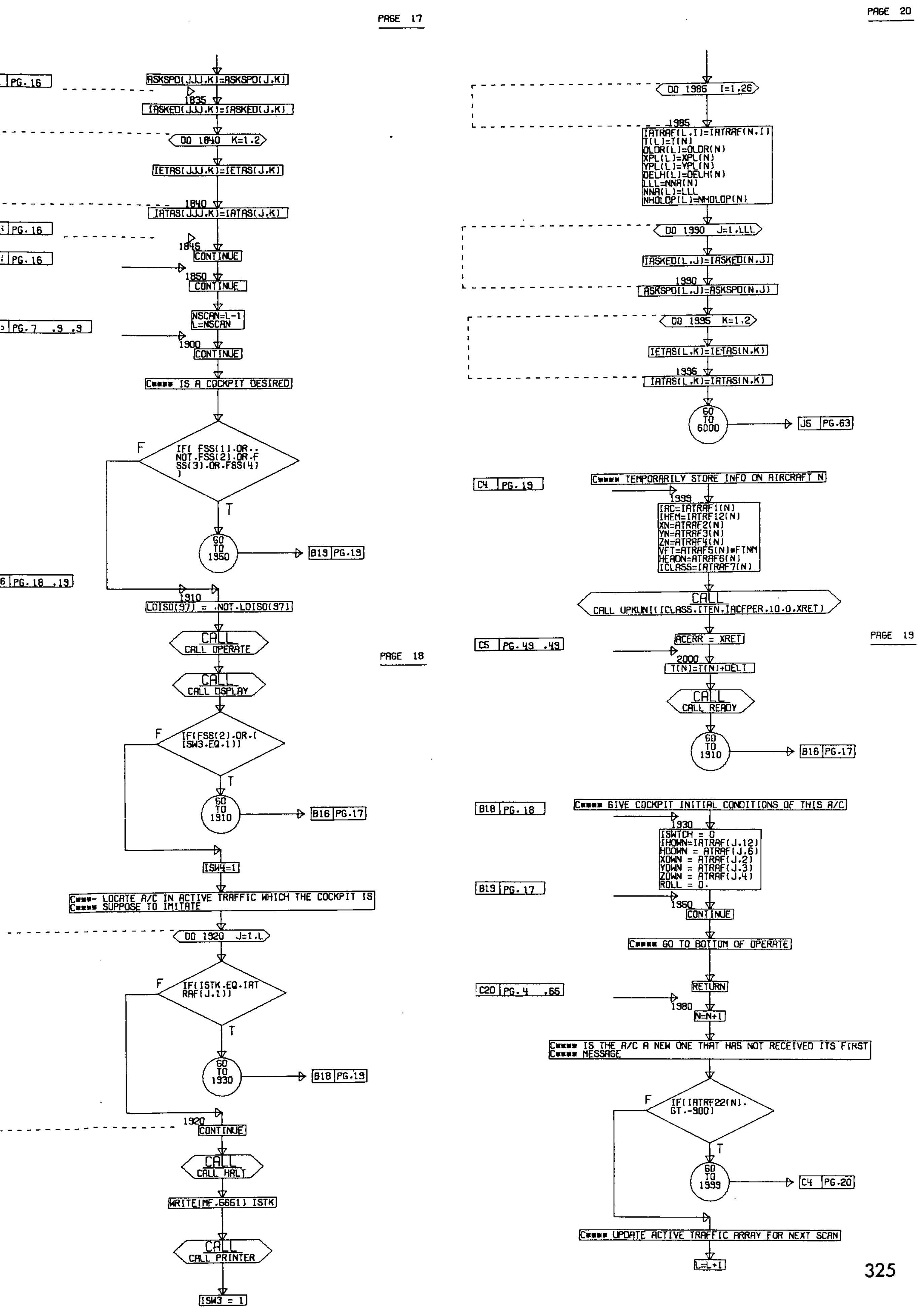
```
*****
```

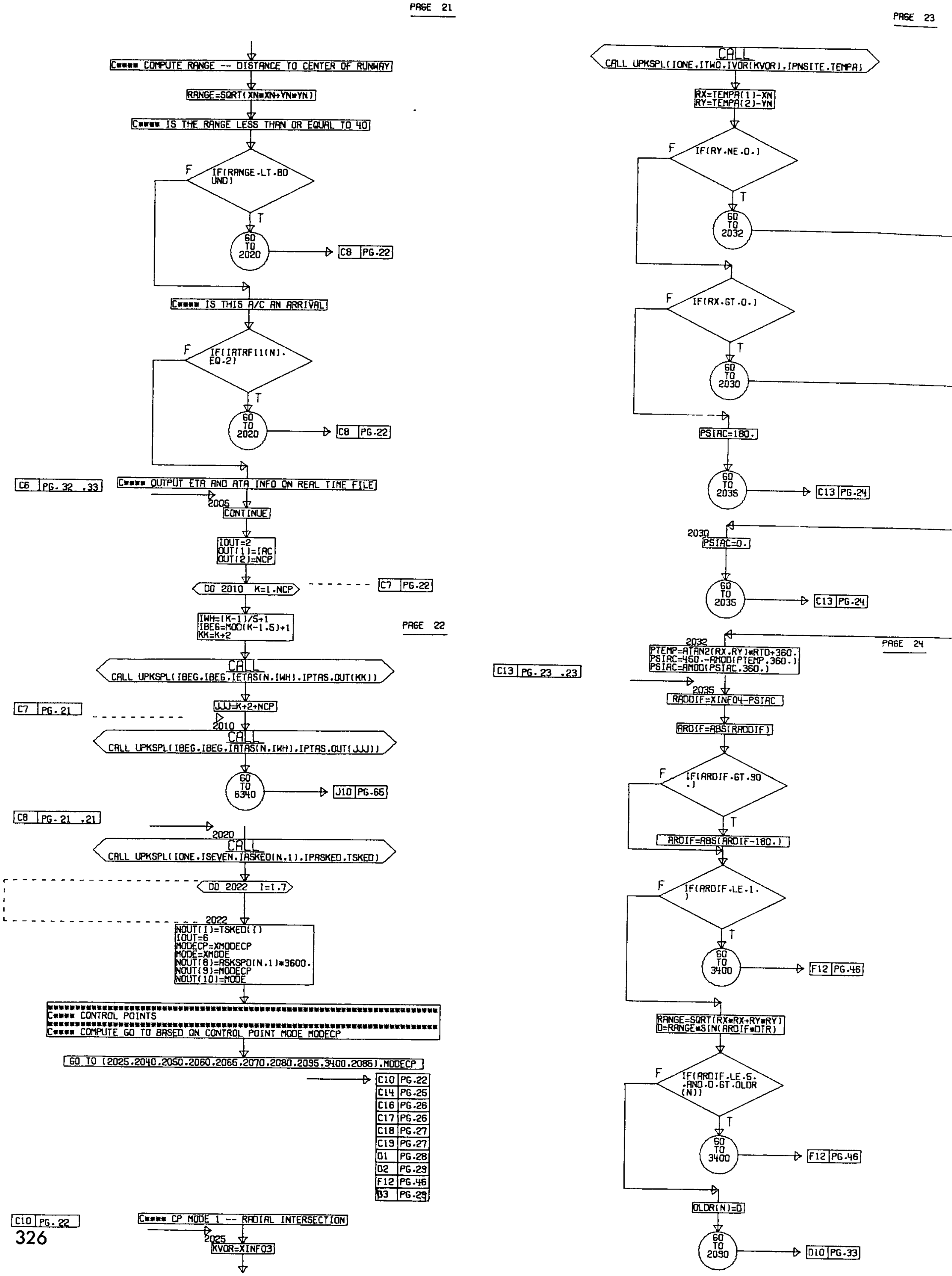
```
75
IF(N.LT.NSCAN)
  T: GO TO 1980  -> C20 PG-19
  F
```

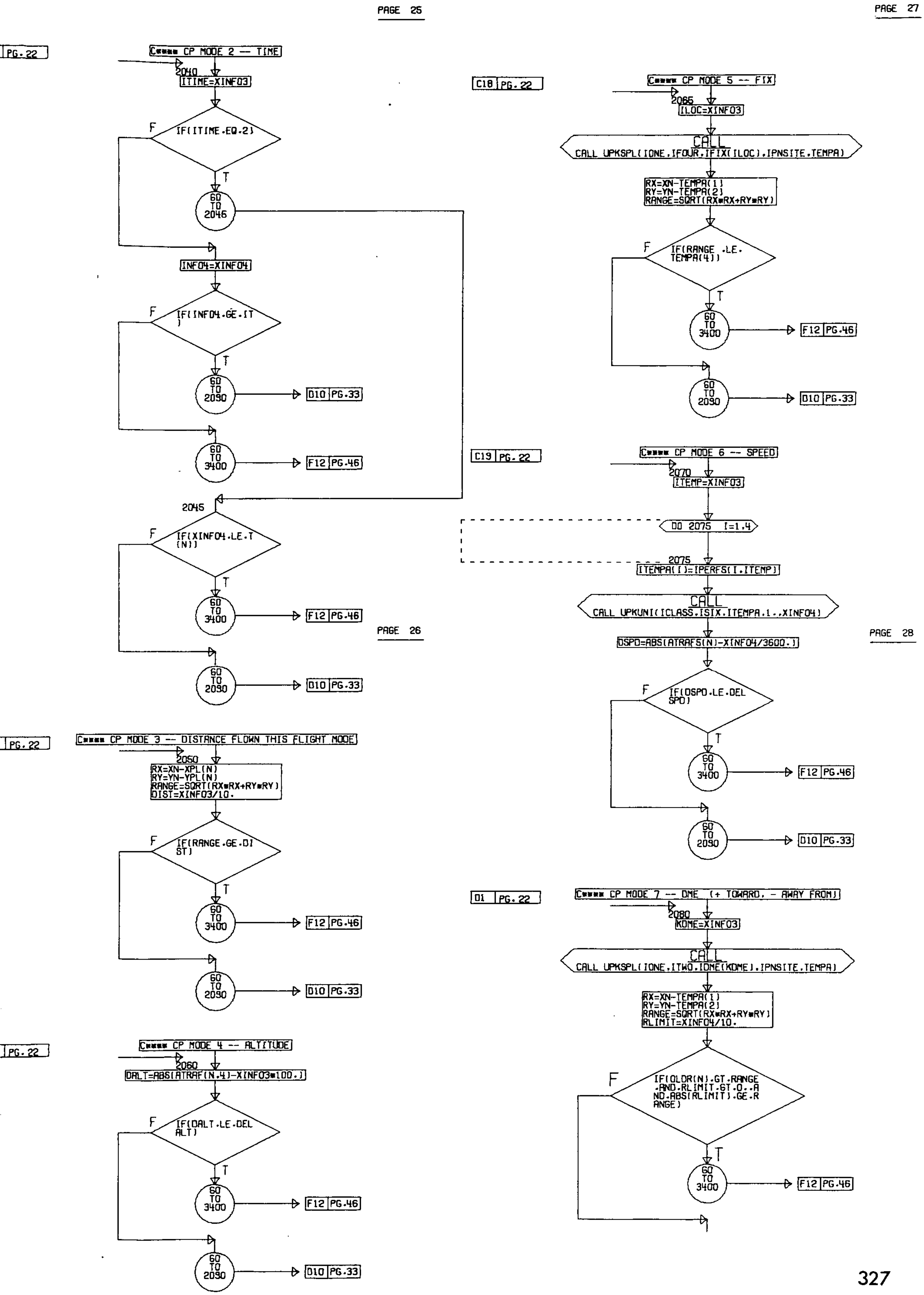

PAGE 6
C**** THIS IS THE FIRST FRAME AFTER THE TRACKING PORTION OF
C**** PROGRAM
C**** WILL A POP-UP BE ADDED TO TRAFFIC THIS SCAN
IF(FSS(1).OR.FSS(2).OR.FSS(3).OR..N.FSS(4))
F
T
GO TO 1150
A11 PG.9
1000
LDISO(99)=.N.LDISO(99)
C**** INPUT INITIAL CONDITIONS FOR POP-UP ON CONSOLE
CALL
CALL OPERATE
CALL
CALL DSPLAY
IF(FSS(4))
F
T
GO TO 1000
POPUP(5)=POPUP(5)/3600.
C**** IS POP-UP AN ARRIVAL
PAGE 6
IF(POPUP(11).EQ.2.)
F
T
GO TO 1020
A6 PG.7
IHEM=IPOPUP(12)
ISAV=ID(IHEM)
IF(ISAV.EQ.0)
F
T
GO TO 1011
C**** PLACE POPUP IN FRONT OF TAKE-OFF QUEUE
DO 1010 J=1,ISAV
JJ=ISAV-J+1
JJJ=JJ+1
1010
ITOQUE(IHEM,JJJ)=ITOQUE(IHEM,JJ)
1011
ID(IHEM)=ID(IHEM)+1
CALL
CALL PAKSPL(IONE,IFOUR,ITRAF1(102),IPTRAF1,POPUP)
DO 1012 J=5,13
A5 PG.7
JJ=J-4
A5 PG.6
1012
TEMPA(JJ)=POPUP(J)
CALL
CALL PAKSPL(IONE,ININE,ITRAF2(102),IPTRAF2,TEMPA)
ITOQUE(IHEM,1)=102
GO TO 1300
B15 PG.17
PAG
A6 PG.6
C**** PLACE POP-UP IN ACTIVE TRAFFIC
1020
NSCAN=L+1
L=NSCAN
IPOPUP(1)=POPUP(1)
IPOPUP(7)=POPUP(7)
IPOPUP(8)=POPUP(8)
IPOPUP(9)=POPUP(9)
DO 1025 K=1,8
1025
ATRAF(NSCAN,K)=POPUP(K)
NHOLDP(NSCAN)=0
T(NSCAN)=0.
OLDR(NSCAN)=80.
XPL(NSCAN)=ATRAF2(NSCAN)
YPL(NSCAN)=ATRAF3(NSCAN)
DELH(NSCAN)=0.
NNA(NSCAN)=1
ICLASS=IATRAF7(NSCAN)
IHEM=1
IF(ATRAF2(NSCAN).LT.0.)
F
T
IHEM=2
IATRAF9(NSCAN)=IPOPUP(9)+NOEPR
IATRF10(NSCAN)=IT
IATRF11(NSCAN)=2
IATRF12(NSCAN)=IHEM
DO 1075 K=13,15
A8 PG.8
A8 PG.7
ATRAF(NSCAN,K)=999.
PAGE
1075
CONTINUE
IATRF16(NSCAN)=0
IATRF17(NSCAN) = 0
ATRF18(NSCAN)=ATRAF2(NSCAN)
ATRF19(NSCAN)=ATRAF3(NSCAN)
ATRF20(NSCAN)=ATRAF4(NSCAN)
DO 1080 K=21,25
1080
IATRAF(NSCAN,K)=0
IATRF22(NSCAN)=1
ATRAF(NSCAN,25)=POPUP(13)
C**** GIVE POPUP AN INITIAL FLIGHT SEGMENT
CALL
CALL UPKSPL(IFOUR,IFOUR,NNPATH(IPOPPA),IPNPATH,TEMP)
IVECT = TEMP
CALL
CALL UPKSPL(IONE,IEIGHT,ISKED(IVECT),IPSKED,TEMPA)
TEMPA(2)=ATRAF6(NSCAN)
TEMPA(4)=ATRAF4(NSCAN)
TEMPA(5)=TEMPA(6)
TEMPA(6)=TEMPA(7)
TEMPA(7)=TEMPA(8)
CALL
CALL PAKSPL(IONE,ISEVEN,IASKED(NSCAN,1),IPASKED,TEMPA)
ASKSPD(NSCAN,1)=POPUP(5)
C**** PLACE A/C IN CONTROLLER ACTION ARRAY
ICA=ICA+1
IACTN1(ICA)=NSCAN
IACTN2(ICA)=6
IACTN3(ICA)=POPUP(13)
C**** INITIALIZE IETAS AND IATAS


PAGE 9
DO 1110 K=1,2
IETAS(NSCAN,K)=0
1110
IATAS(NSCAN,K)=0
GO TO 1900
B15 PG.17
1 PG.5
C**** IS THERE A A/C THAT NEEDS TO BE DELETED
1150
CONTINUE
IF(FSS(1).OR.FSS(2).OR..NOT.FSS(3).OR.FSS(4))
F
T
GO TO 1900
B15 PG.17
PG.10 .11
1600
LDISO(98)=.NOT.LDISO(98)
CALL
CALL OPERATE
CALL
CALL DSPLAY
IF(FSS(3).OR.(ISW2.EQ.1))
F
TOP OF NEXT PAGE
T
GO TO 1600
A12 PG.9
PAGE 10
BOTTOM OF PRECEDING PAGE
LDISO(98)=.F.
C**** LOCATE A/C TO BE DELETED ON ID IN ACTIVE TRAFFIC
DO 1610 IDD1=1,L
IF(IDELET.NE.IATRAF1(IDD1))
F
T
GO TO 1610
IAC=IDD1
GO TO 1611
A14 PG.11
1610
CONTINUE
CALL
CALL HALT
WRITE(NF,5649) IDELET
CALL
CALL PRINTER
CALL
CALL READY
PAGE 11
ISW2=1
GO TO 1600
A12 PG.9
A14 PG.10
1611
IDENT=IDELET
GO TO 1613
1613
LL = IATRAF9(IAC)
C**** UNPACK SUBSCRIPT OF ETA AT THE END OF THE PRESENT PATH
CALL
CALL UPKSPL(IFIVE,IFIVE,IATRF21(IAC),IPRPATH,TEMP)
ISUB =TEMP
ASSIGN 1660 TO NGO
IF(ISUB .EQ.0)
F
T
GO TO 1635
A18 PG.12
PAGE 12
JTEMP=NETA(ISUB)
IF(JTEMP.LE.0)
F
T
GO TO 1635
A16 PG.14
C**** LOCATE A/C IN ETA ARRAY
1625
DO 1630 J=1,JTEMP
CALL
CALL UPKSPL(IONE,IONE,IETA1(ISUB ,J),IPETA1,TEMP)
ITEMP=TEMP
IF(ITEMP.EQ.IDENT)
F
T
GO TO 1640
B20 PG.13
1630
CONTINUE
C**** WAS UNABLE TO LOCATE A/C IN ETA ARRAY
A18 PG.11
1635
CALL
CALL HALT
1636
ENCODE(107,5650,TYBUF) ISUB,IDENT,IT

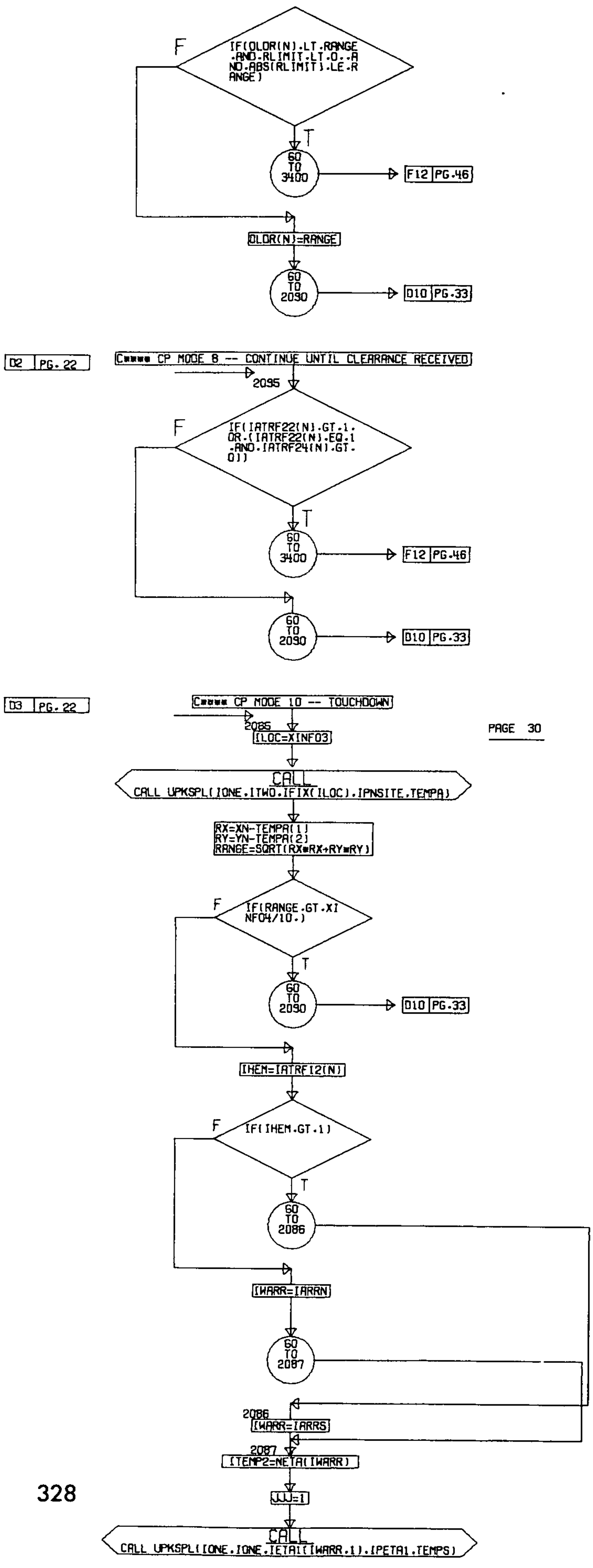

PAGE 13
IBY=0
CALL
CALL WRT218(TYBUF,IBY,IXI)
CALL
CALL READY
GO TO NGO (1660,1675)
B3 PG.14
B4 PG.14
C**** DELETE A/C FROM ETA ARRAY AT END OF PRESENT PATH
C**** MOVE ALL OTHER A/C TOWARD THE FRONT
B20 PG.12
1640
JJ=J+1
IF(JJ.GT.JTEMP)
F
T
GO TO 1650
DO 1645 J=JJ,JTEMP
JJJ=J-1
IETA1(ISUB,JJJ)=IETA1(ISUB,J)
1645
IETA2(ISUB,JJJ)=IETA2(ISUB,J)
1650
NETA(ISUB)=JTEMP-1
GO TO NGO (1660,1675)
B3 PG.14
B4 PG.14
PAGE 14
C**** UNPACK SUBSCRIPT OF ETA AT END OF THE FUTURE PATH
B3 PG.13 ,13
1660
CALL
CALL UPKSPL(IFIVE,IFIVE,IATRF23(IAC),IPAPATH,TEMP)
ISUB=TEMP
ASSIGN 1675 TO NGO
IF(ISUB.EQ.0)
F
T
GO TO 1675
JTEMP=NETA(ISUB)
IF(JTEMP.LE.0) 1675,1625
T
F
A16 PG.12
B4 PG.13 ,13
1675
CONTINUE
C**** DELETE A/C FROM SEQUENCING QUEUE IF IT HAS BEEN SEQUENCED
IF(ISEQ.EQ.0)
F
T
GO TO 1800
B9 PG.16
PAGE 15
C**** CHECK IDS OF A/C IN SEQUENCING QUEUE
DO 1700 J=1,ISEQ
IF(ISEQUE(J,1) .EQ. IDENT)
F
T
GO TO 1720
1700
CONTINUE
GO TO 1800
B9 PG.16
C**** DELETE A/C FROM SEQUENCING QUEUE
1720
JJ=J+1
IF(JJ.GT.ISEQ)
F
T
GO TO 1730
B8 PG.16
DO 1725 J=JJ,ISEQ
B7 PG.16
JJJ=J-1
DO 1725 K=1,16
B7 PG.16
PAGE 16
B7 PG.15 ,15
1725
ISEQUE(JJJ,K)=ISEQUE(J,K)
B8 PG.15
1730
ISEQ=ISEQ-1
B9 PG.14 ,15
1800
CONTINUE
C**** DELETE A/C FROM THE ACTIVE TRAFFIC SAMPLE
IF(IAC.GE.L)
F
T
GO TO 1850
B14 PG.17
JJ=IAC+1
C**** MOVE ALL OTHER A/C DOWN IN ACTIVE TRAFFIC
DO 1845 J=JJ,L
B13 PG.17
JJJ=J-1
DO 1825 K=1,26
1825
ATRAF(JJJ,K)=ATRAF(J,K)
NHOLDP(JJJ)=NHOLDP(J)
T(JJJ)=T(J)
OLDR(JJJ)=OLDR(J)
XPL(JJJ)=XPL(J)
YPL(JJJ)=YPL(J)
DELH(JJJ)=DELH(J)
LLL=NNA(J)
NNA(JJJ)=LLL
DO 1835 K=1,LLL
B11 PG.17


PAGE 17
PG. 16
ASKSPD(JJJ,K)=ASKSPD(J,K)
1835
IASKED(JJJ,K)=IASKED(J,K)
DO 1840 K=1,2
IETAS(JJJ,K)=IETAS(J,K)
1840
IATAS(JJJ,K)=IATAS(J,K)
PG. 16
1845
CONTINUE
PG. 16
1850
CONTINUE
NSCAN=L-1
L=NSCAN
PG. 7 ,9 ,9
1900
CONTINUE
C**** IS A COCKPIT DESIRED
IF(FSS(1).OR.,
NOT.FSS(2).OR.F
SS(3).OR.FSS(4)
)
F
T
GO TO 1950
B19 PG.19
PG. 18 ,19
1910
LDISO(97) = .NOT.LDISO(97)
CALL
CALL OPERATE
CALL
CALL DSPLAY
IF(FSS(2).OR.(
ISW3.EQ.1))
F
T
GO TO 1910
B16 PG.17
ISW4=1
C***- LOCATE A/C IN ACTIVE TRAFFIC WHICH THE COCKPIT IS
C**** SUPPOSE TO IMITATE
DO 1920 J=1,L
IF(ISTK.EQ.IAT
RAF(J,1))
F
T
GO TO 1930
B18 PG.19
1920
CONTINUE
CALL
CALL HALT
WRITE(MF,6651) ISTK
CALL
CALL PRINTER
ISW3 = 1
PAGE 18
PAGE 20
DO 1985 I=1,26
1985
IATRAF(L,I)=IATRAF(N,I)
T(L)=T(N)
OLDR(L)=OLDR(N)
XPL(L)=XPL(N)
YPL(L)=YPL(N)
DELH(L)=DELH(N)
LLL=NNA(N)
NNA(L)=LLL
NHOLDP(L)=NHOLDP(N)
DO 1990 J=1,LLL
IASKED(L,J)=IASKED(N,J)
1990
ASKSPD(L,J)=ASKSPD(N,J)
DO 1995 K=1,2
IETAS(L,K)=IETAS(N,K)
1995
IATAS(L,K)=IATAS(N,K)
GO TO 6000
J5 PG.63
C4 PG. 19
C**** TEMPORARILY STORE INFO ON AIRCRAFT N
1999
IAC=IATRAF1(N)
IHEM=IATRF12(N)
XN=ATRAF2(N)
YN=ATRAF3(N)
ZN=ATRAF4(N)
VFT=ATRAF5(N)*FTNM
HEADN=ATRAF6(N)
ICLASS=IATRAF7(N)
CALL
CALL UPKUNI(ICLASS,ITEN,IACFPER,10,0,XRET)
C5 PG. 49 ,49
ACERR = XRET
2000
T(N)=T(N)+DELT
CALL
CALL READY
GO TO 1910
B16 PG.17
PAGE 19
B18 PG. 18
C**** GIVE COCKPIT INITIAL CONDITIONS OF THIS A/C
1930
ISWTCH = 0
IHOWN=IATRAF(J,12)
HDOWN = ATRAF(J,6)
XOWN = ATRAF(J,2)
YOWN = ATRAF(J,3)
ZOWN = ATRAF(J,4)
ROLL = 0.
B19 PG. 17
1950
CONTINUE
C**** GO TO BOTTOM OF OPERATE
C20 PG. 4 ,65
RETURN
1980
N=N+1
C**** IS THE A/C A NEW ONE THAT HAS NOT RECEIVED ITS FIRST
C**** MESSAGE
IF(IATRF22(N).
GT.-900)
F
T
GO TO 1999
C4 PG.20
C**** UPDATE ACTIVE TRAFFIC ARRAY FOR NEXT SCAN
L=L+1

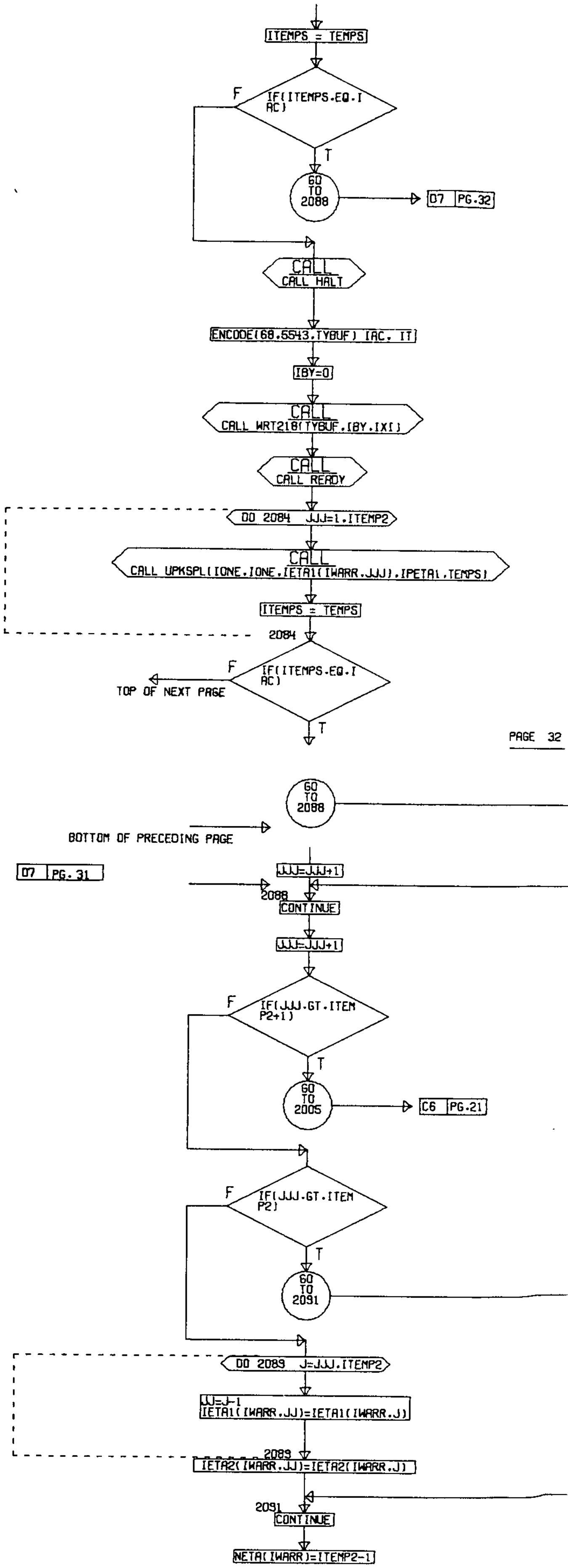

PAGE 21
C**** COMPUTE RANGE -- DISTANCE TO CENTER OF RUNWAY
RANGE=SQRT(XN*XN+YN*YN)
C**** IS THE RANGE LESS THAN OR EQUAL TO 40
IF(RANGE.LT.BOUND)
F
T
GO TO 2020
C8 PG.22
C**** IS THIS A/C AN ARRIVAL
IF(IATRF11(N).EQ.2)
F
T
GO TO 2020
C8 PG.22
C6 PG.32 .33
C**** OUTPUT ETA AND ATA INFO ON REAL TIME FILE
2006
CONTINUE
IOUT=2
OUT(1)=IAC
OUT(2)=NCP
DO 2010 K=1,NCP
C7 PG.22
PAGE 22
IWH=(K-1)/5+1
IBEG=MOD(K-1,5)+1
KK=K+2
CALL
CALL UPKSPL(IBEG,IBEG,IETAS(N,IWH),IPTAS,OUT(KK))
C7 PG.21
JJJ=K+2+NCP
2010
CALL
CALL UPKSPL(IBEG,IBEG,IATAS(N,IWH),IPTAS,OUT(JJJ))
GO TO 6340
J10 PG.65
C8 PG.21 .21
2020
CALL
CALL UPKSPL(IONE,ISEVEN,IASKED(N,1),IPASKED,TSKED)
DO 2022 I=1,7
2022
NOUT(I)=TSKED(I)
IOUT=6
MODECP=XMODECP
MODE=XMODE
NOUT(8)=ASKSPD(N,1)*3600.
NOUT(9)=MODECP
NOUT(10)=MODE
C**** CONTROL POINTS
C**** COMPUTE GO TO BASED ON CONTROL POINT MODE MODECP
GO TO (2025,2040,2050,2060,2065,2070,2080,2095,3400,2085),MODECP
C10 PG.22
C14 PG.25
C16 PG.26
C17 PG.26
C18 PG.27
C19 PG.27
D1 PG.28
D2 PG.29
F12 PG.46
D3 PG.29
C10 PG.22
C**** CP MODE 1 -- RADIAL INTERSECTION
2025
KVOR=XINFO3
C13 PG.23 .23
PAGE 23
CALL
CALL UPKSPL(IONE,ITWO,IVOR(KVOR),IPNSITE,TEMPA)
RX=TEMPA(1)-XN
RY=TEMPA(2)-YN
IF(RY.NE.0.)
F
T
GO TO 2032
IF(RX.GT.0.)
F
T
GO TO 2030
PSIAC=180.
GO TO 2035
C13 PG.24
2030
PSIAC=0.
GO TO 2035
C13 PG.24
PAGE 24
2032
PTEMP=ATAN2(RX,RY)*RTD+360.
PSIAC=450.-AMOD(PTEMP,360.)
PSIAC=AMOD(PSIAC,360.)
2035
RADDIF=XINFO4-PSIAC
ARDIF=ABS(RADDIF)
IF(ARDIF.GT.90.)
F
T
ARDIF=ABS(ARDIF-180.)
IF(ARDIF.LE.1.)
F
T
GO TO 3400
F12 PG.46
RANGE=SQRT(RX*RX+RY*RY)
D=RANGE*SIN(ARDIF*DTR)
IF(ARDIF.LE.5..AND.D.GT.OLDR(N))
F
T
GO TO 3400
F12 PG.46
OLDR(N)=D
GO TO 2090
D10 PG.33

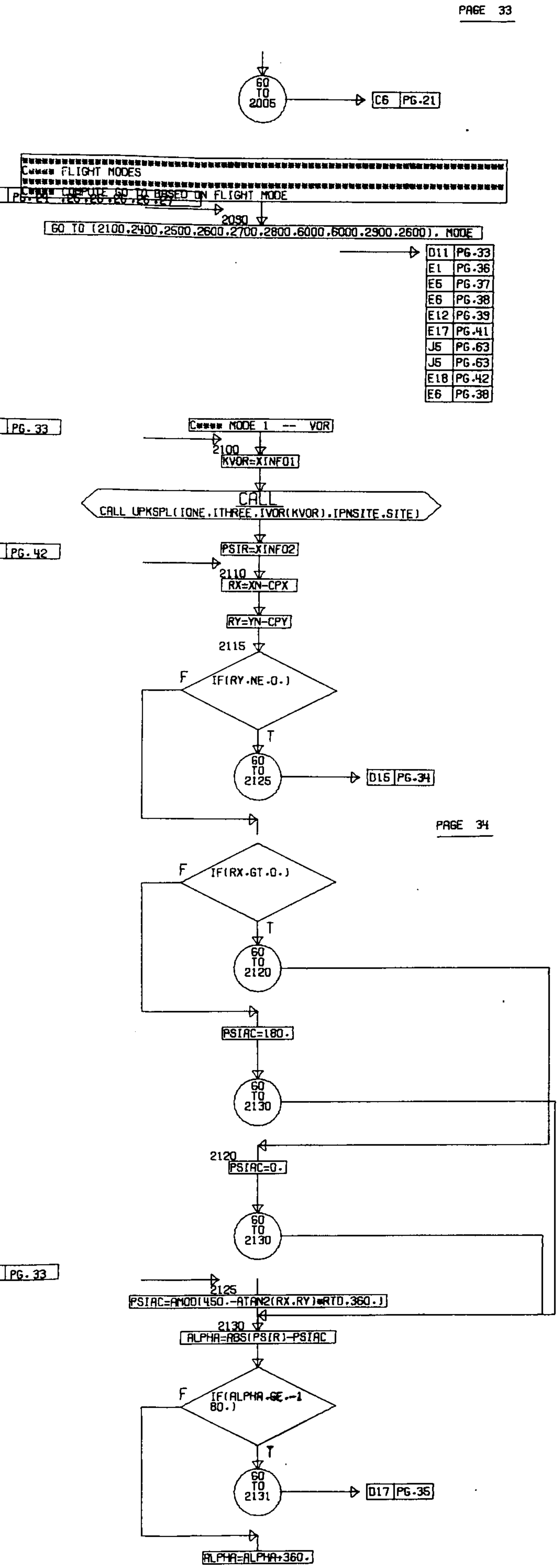

PAGE 25
PG. 22
C**** CP MODE 2 — TIME
2040
ITIME=XINFO3
IF(ITIME.EQ.2)
F
T
GO TO 2046
INFO4=XINFO4
IF(INFO4.GE.IT)
GO TO 2090
D10 PG.33
GO TO 3400
F12 PG.46
2045
IF(XINFO4.LE.T(N))
GO TO 3400
F12 PG.46
GO TO 2090
D10 PG.33
PAGE 26
PG. 22
C**** CP MODE 3 -- DISTANCE FLOWN THIS FLIGHT MODE
2050
RX=XN-XPL(N)
RY=YN-YPL(N)
RANGE=SQRT(RX*RX+RY*RY)
DIST=XINFO3/10.
IF(RANGE.GE.DIST)
GO TO 3400
F12 PG.46
GO TO 2090
D10 PG.33
PG. 22
C**** CP MODE 4 -- ALTITUDE
2060
DALT=ABS(ATRAF(N.4)-XINFO3*100.)
IF(DALT.LE.DELALT)
GO TO 3400
F12 PG.46
GO TO 2090
D10 PG.33
PAGE 27
C18 PG. 22
C**** CP MODE 5 -- FIX
2065
ILOC=XINFO3
CALL
CALL UPKSPL(IONE.IFOUR.IFIX(ILOC).IPNSITE.TEMPA)
RX=XN-TEMPA(1)
RY=YN-TEMPA(2)
RANGE=SQRT(RX*RX+RY*RY)
IF(RANGE .LE. TEMPA(4))
GO TO 3400
F12 PG.46
GO TO 2090
D10 PG.33
C19 PG. 22
C**** CP MODE 6 -- SPEED
2070
ITEMP=XINFO3
DO 2075 I=1.4
2075
ITEMPA(I)=IPERFS(I.ITEMP)
CALL
CALL UPKUNI(ICLASS.ISIX.ITEMPA.1..XINFO4)
DSPD=ABS(ATRAFS(N)-XINFO4/3600.)
IF(DSPD.LE.DELSPD)
GO TO 3400
F12 PG.46
GO TO 2090
D10 PG.33
PAGE 28
D1 PG. 22
C**** CP MODE 7 -- DME (+ TOWARD, - AWAY FROM)
2080
KDME=XINFO3
CALL
CALL UPKSPL(IONE.ITWO.IDME(KDME).IPNSITE.TEMPA)
RX=XN-TEMPA(1)
RY=YN-TEMPA(2)
RANGE=SQRT(RX*RX+RY*RY)
RLIMIT=XINFO4/10.
IF(OLDR(N).GT.RANGE.AND.RLIMIT.GT.O..AND.ABS(RLIMIT).GE.RANGE)
GO TO 3400
F12 PG.46

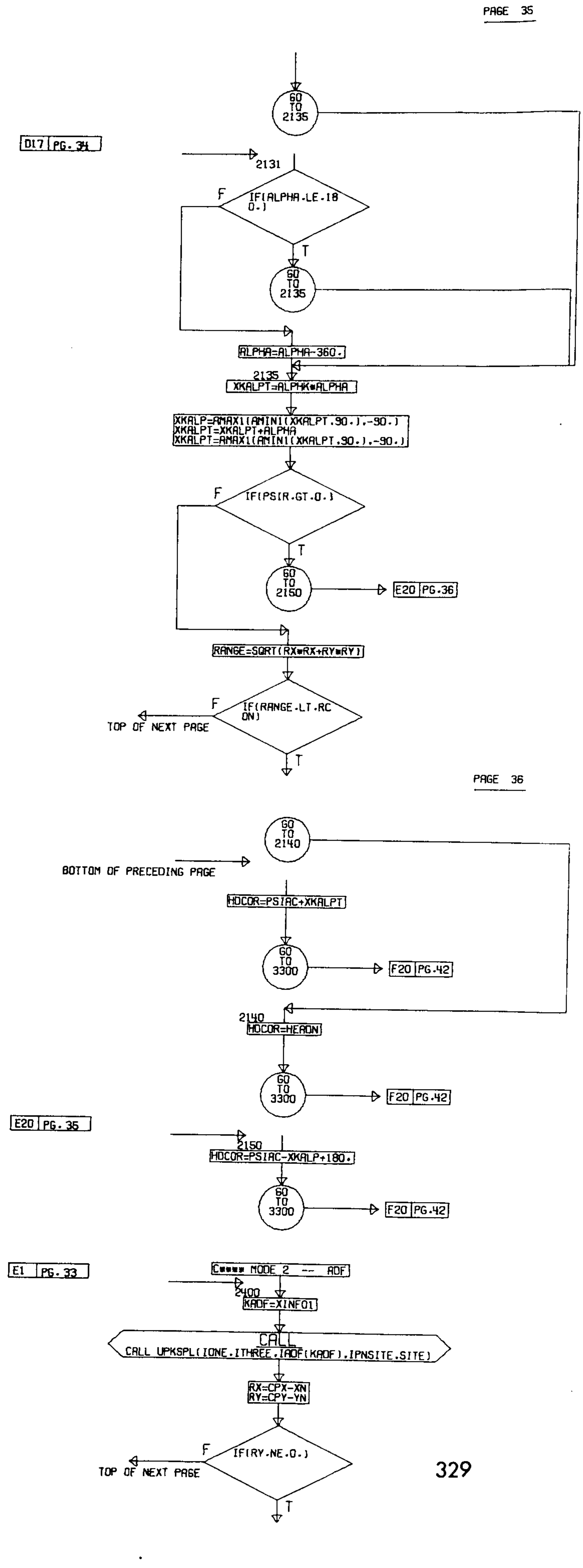
PAGE 29



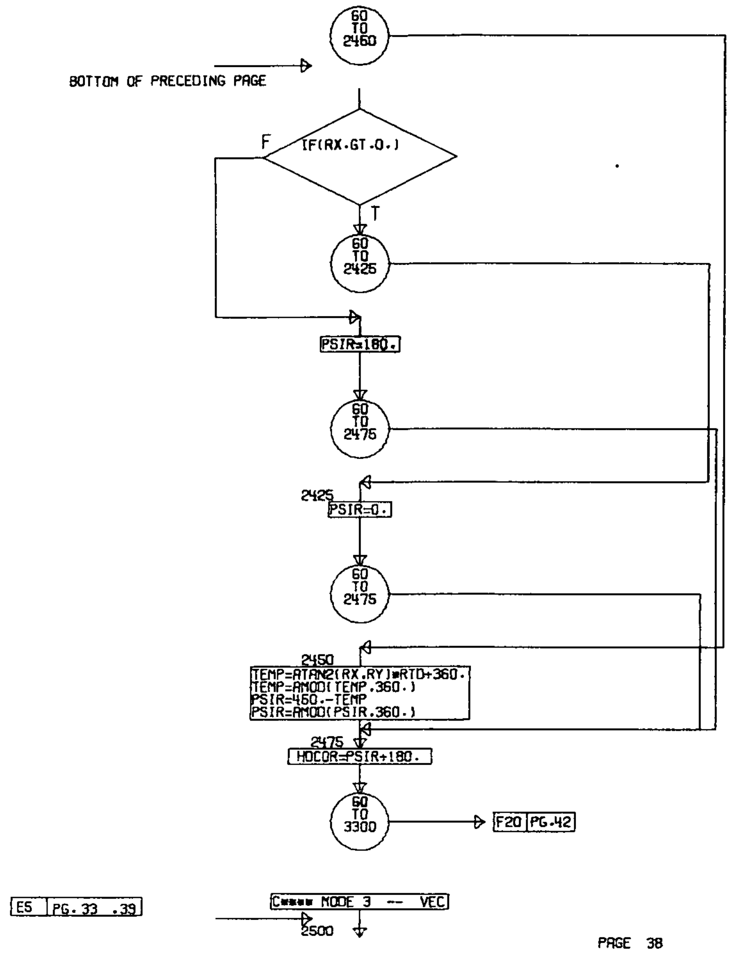
PAGE 31



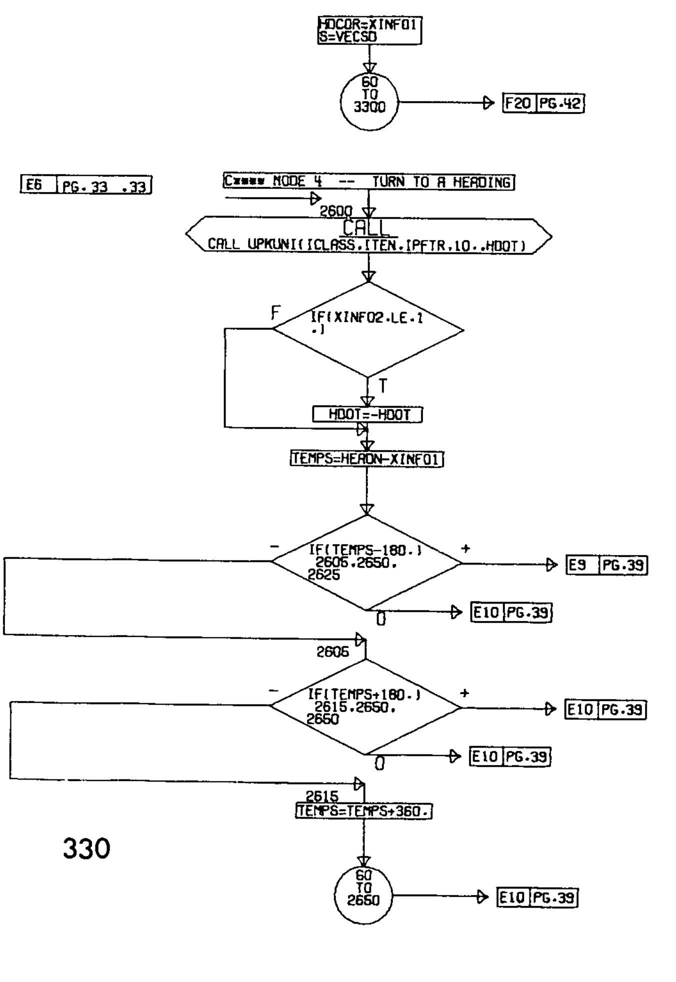

PAGE 33
GO TO 2005
C6 PG.21
C**** FLIGHT MODES
C**** COMPUTE GO TO BASED ON FLIGHT MODE
PG.21 .26.26.26.26.27
2090
GO TO (2100,2400,2500,2600,2700,2800,6000,6000,2900,2600), MODE
D11 PG.33
E1 PG.36
E6 PG.37
E6 PG.38
E12 PG.39
E17 PG.41
J5 PG.63
J5 PG.63
E18 PG.42
E6 PG.38
PG. 33
C**** MODE 1 -- VOR
2100
KVOR=XINFO1
CALL
CALL UPKSPL(IONE,ITHREE,IVOR(KVOR),IPNSITE,SITE)
PG. 42
PSIR=XINFO2
2110
RX=XN-CPX
RY=YN-CPY
2115
IF(RY.NE.0.)
F
T
GO TO 2125
D16 PG.34
PAGE 34
IF(RX.GT.0.)
F
T
GO TO 2120
PSIAC=180.
GO TO 2130
2120
PSIAC=0.
GO TO 2130
PG. 33
2125
PSIAC=AMOD(450.-ATAN2(RX,RY)*RTD,360.)
2130
ALPHA=ABS(PSIR)-PSIAC
IF(ALPHA.GE.-180.)
F
T
GO TO 2131
D17 PG.35
ALPHA=ALPHA+360.

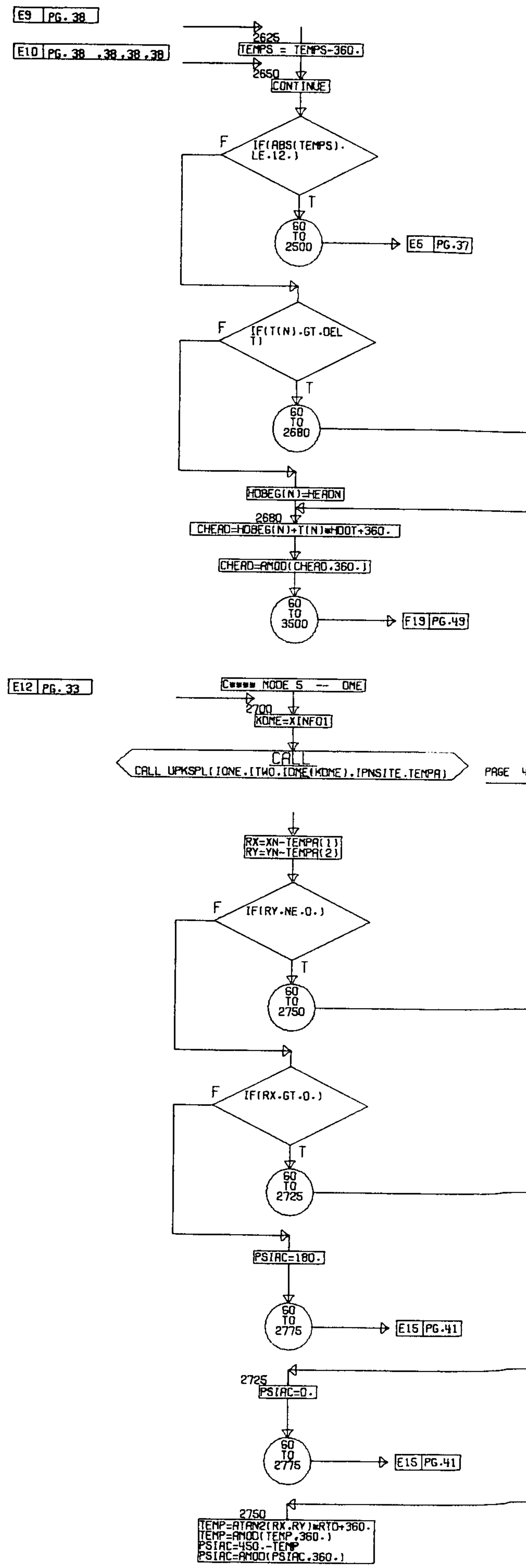

PAGE 35
GO TO 2135
D17 PG. 34
2131
IF(ALPHA.LE.180.)
F
T
GO TO 2135
ALPHA=ALPHA-360.
2135
XKALPT=ALPHK*ALPHA
XKALP=AMAX1(AMIN1(XKALPT,90.),-90.)
XKALPT=XKALPT+ALPHA
XKALPT=AMAX1(AMIN1(XKALPT,90.),-90.)
IF(PSIR.GT.0.)
F
T
GO TO 2150
E20 PG.36
RANGE=SQRT(RX*RX+RY*RY)
IF(RANGE.LT.RCON)
F
TOP OF NEXT PAGE
T
PAGE 36
GO TO 2140
BOTTOM OF PRECEDING PAGE
HDCOR=PSIAC+XKALPT
GO TO 3300
F20 PG.42
2140
HDCOR=HEADN
GO TO 3300
F20 PG.42
E20 PG. 35
2150
HDCOR=PSIAC-XKALP+180.
GO TO 3300
F20 PG.42
E1 PG. 33
C**** MODE 2 -- ADF
2400
KADF=XINFO1
CALL
CALL UPKSPL(IONE,ITHREE,IADF(KADF),IPNSITE,SITE)
RX=CPX-XN
RY=CPY-YN
IF(RY.NE.0.)
F
TOP OF NEXT PAGE
T

PAGE 37

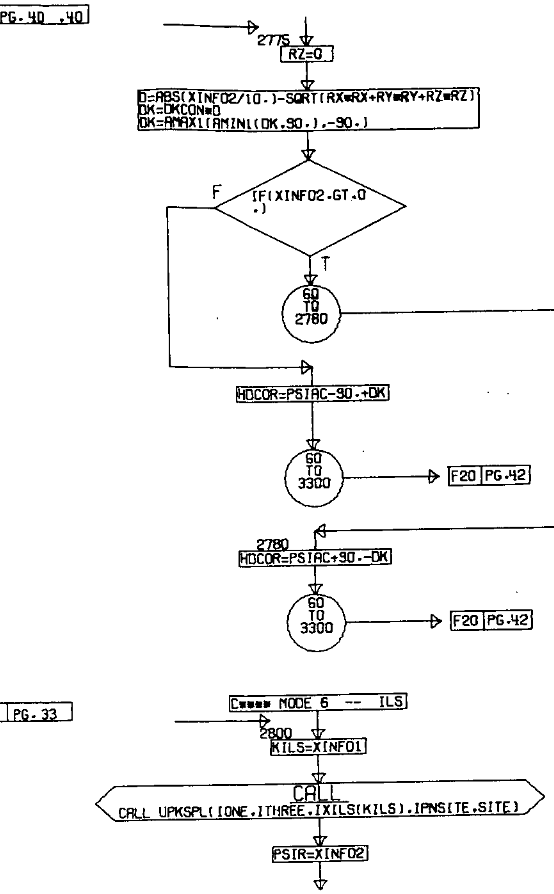


PAGE 38

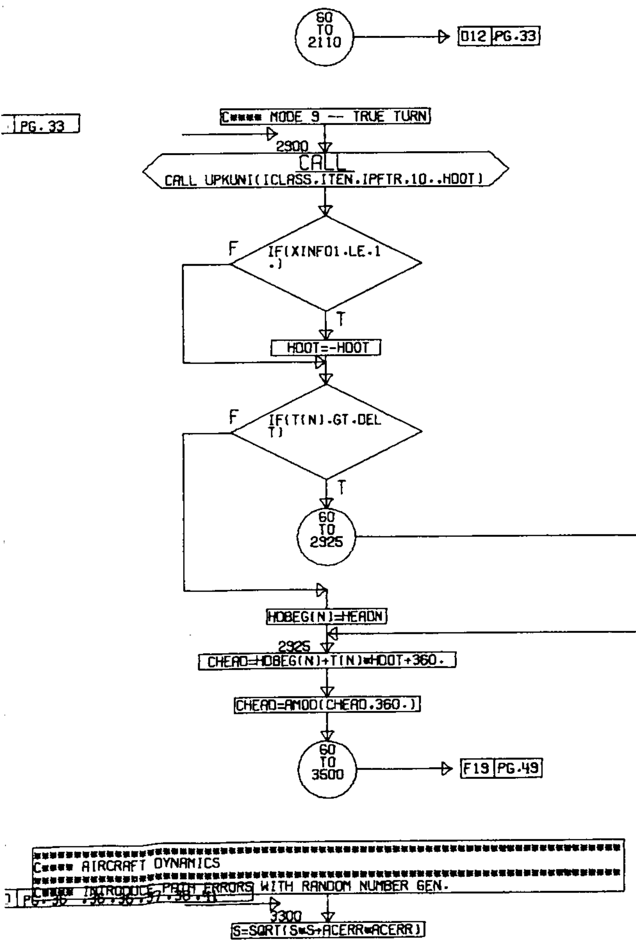




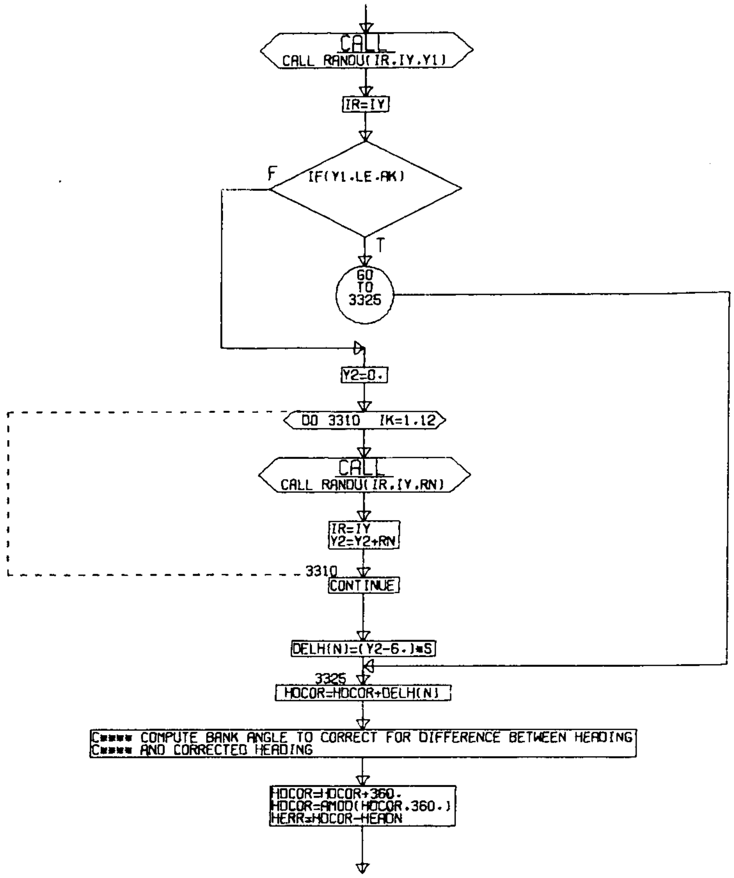
PAGE 41



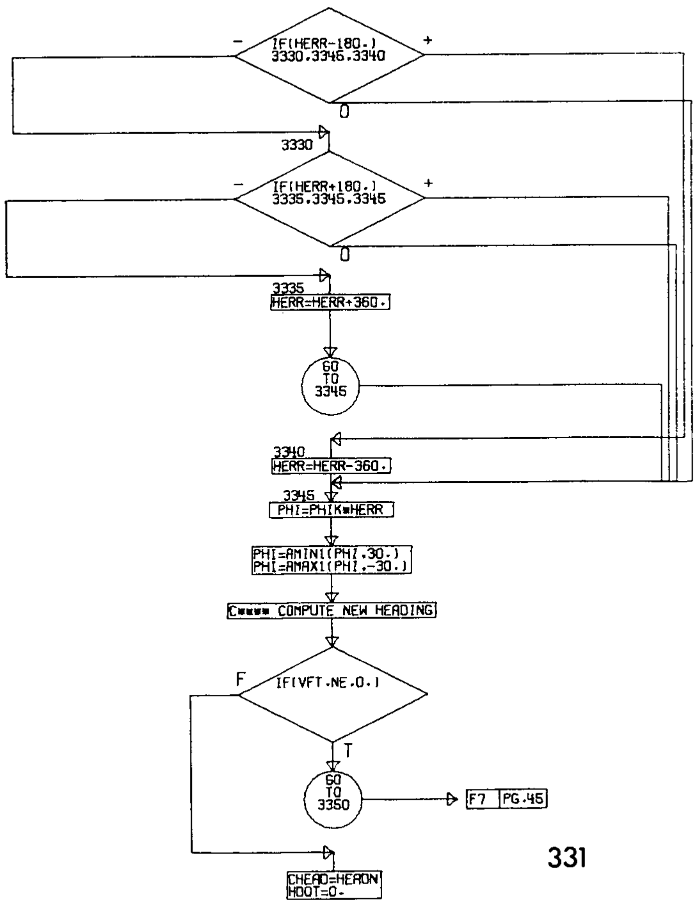
PAGE 42

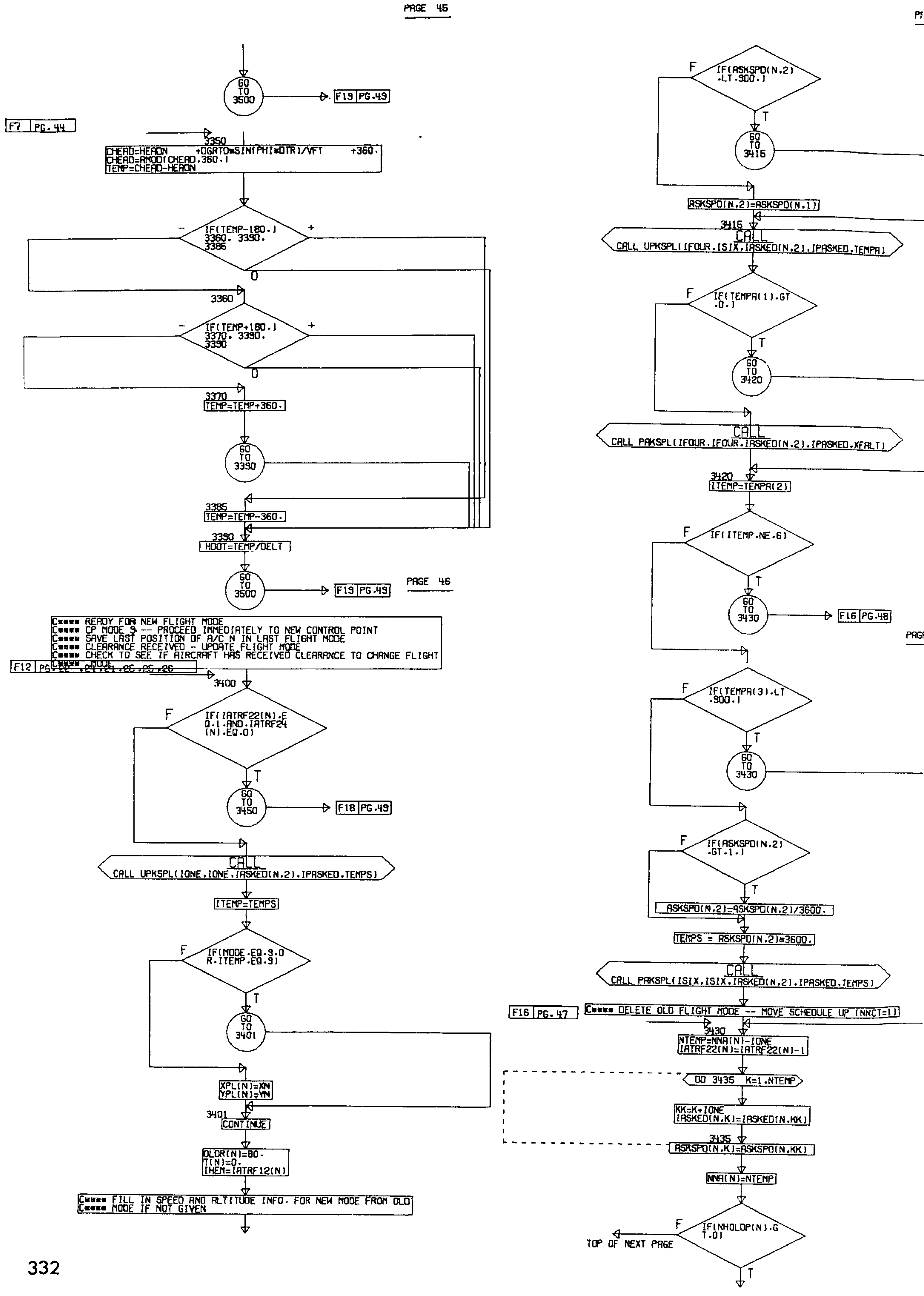


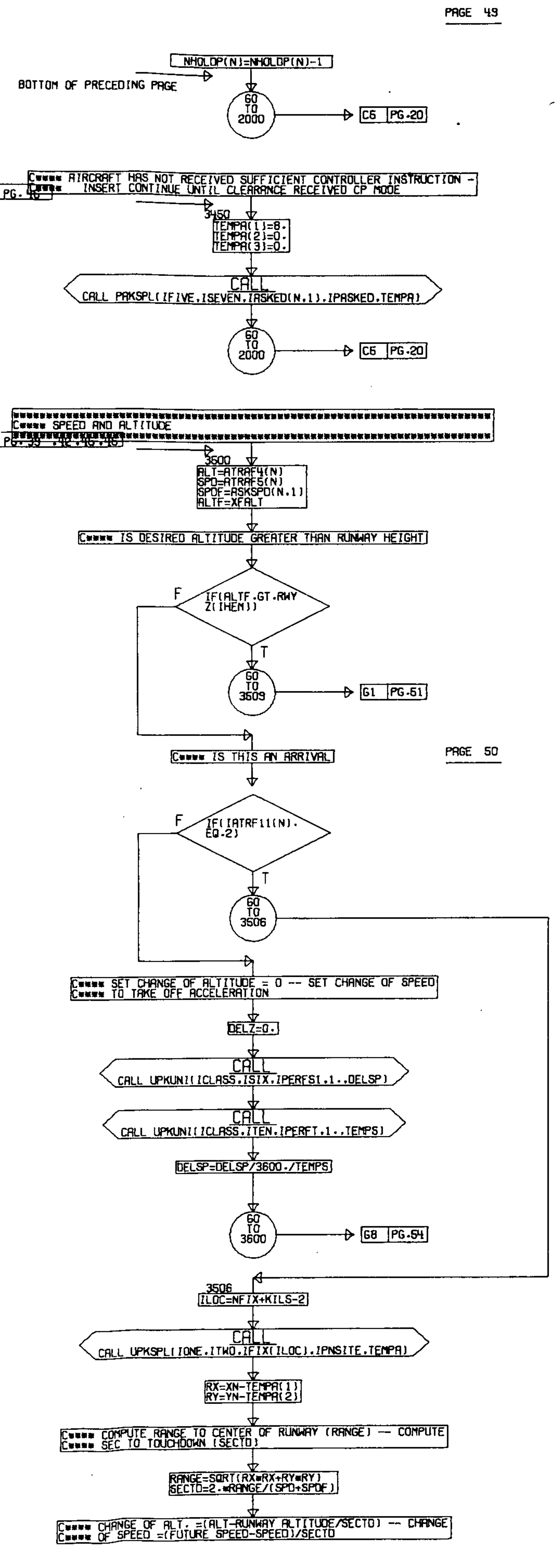
PAGE 43

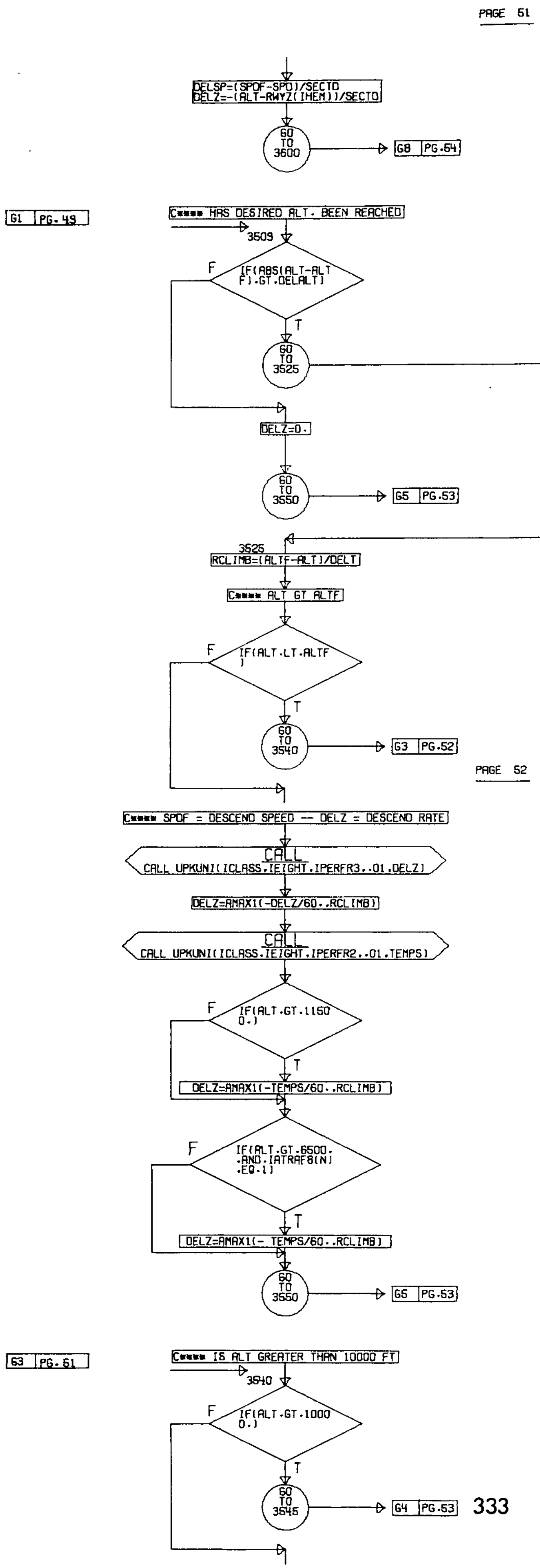


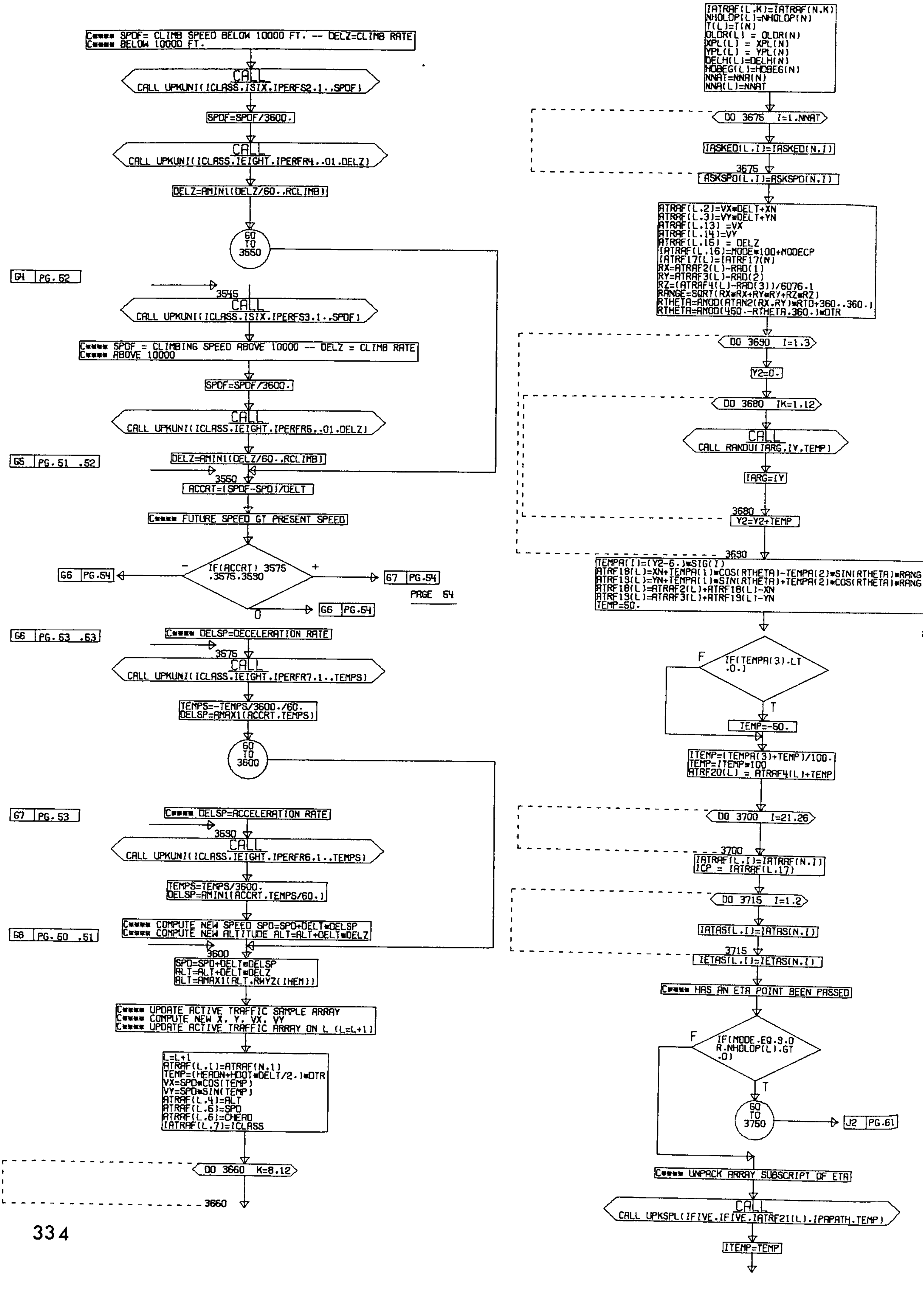
PAGE 44

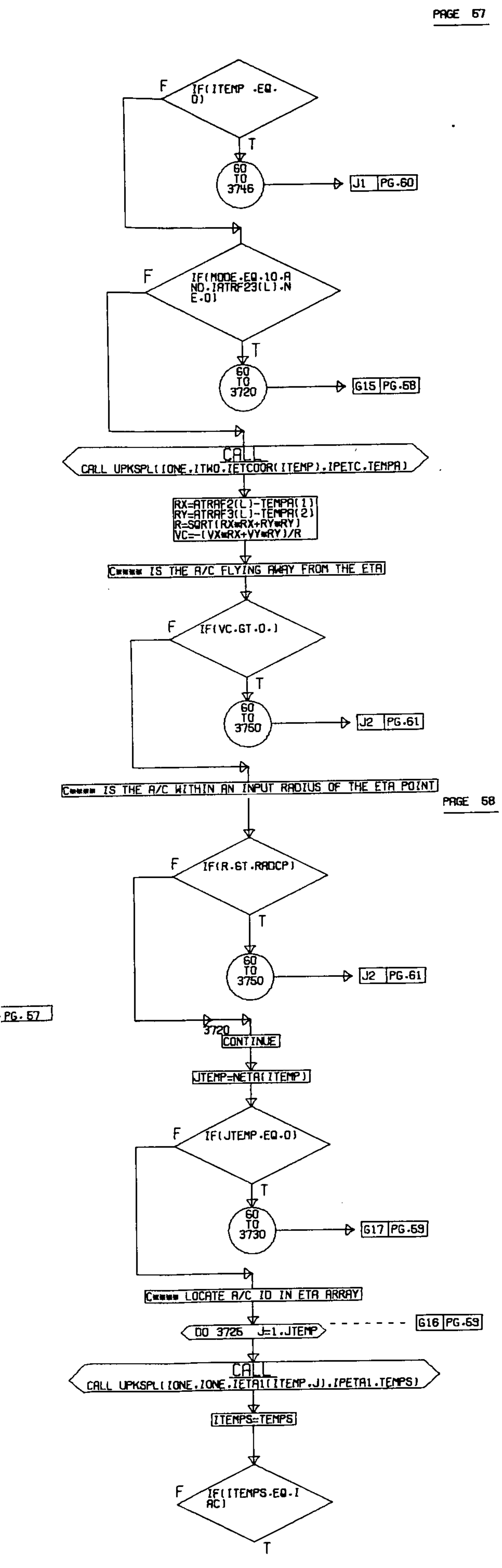



PAGE 45
GO TO 3500
F19 PG.49
F7 PG.44
3350
CHEAD=HEADN +DGRTD*SIN(PHI*DTR)/VFT +360.
CHEAD=AMOD(CHEAD,360.)
TEMP=CHEAD-HEADN
IF(TEMP-180.) 3360, 3390, 3385
3360
IF(TEMP+180.) 3370, 3390, 3390
3370
TEMP=TEMP+360.
GO TO 3390
3385
TEMP=TEMP-360.
3390
HDOT=TEMP/DELT
GO TO 3500
F19 PG.49
PAGE 46
C**** READY FOR NEW FLIGHT MODE
C**** CP MODE 9 -- PROCEED IMMEDIATELY TO NEW CONTROL POINT
C**** SAVE LAST POSITION OF A/C N IN LAST FLIGHT MODE
C**** CLEARANCE RECEIVED - UPDATE FLIGHT MODE
C**** CHECK TO SEE IF AIRCRAFT HAS RECEIVED CLEARANCE TO CHANGE FLIGHT
C**** MODE
F12 PG.22 ,24,24,26,25,26
3400
IF(IATRF22(N).EQ.1.AND.IATRF24(N).EQ.0)
GO TO 3450
F18 PG.49
CALL
CALL UPKSPL(IONE,IONE,IASKED(N,2),IPASKED,TEMPS)
ITEMP=TEMPS
IF(MODE.EQ.9.OR.ITEMP.EQ.9)
GO TO 3401
XPL(N)=XN
YPL(N)=YN
3401
CONTINUE
OLDR(N)=80.
T(N)=0.
IHEN=IATRF12(N)
C**** FILL IN SPEED AND ALTITUDE INFO. FOR NEW MODE FROM OLD
C**** MODE IF NOT GIVEN
IF(ASKSPD(N,2).LT.900.)
GO TO 3416
ASKSPD(N,2)=ASKSPD(N,1)
3415
CALL
CALL UPKSPL(IFOUR,ISIX,IASKED(N,2),IPASKED,TEMPA)
IF(TEMPA(1).GT.0.)
GO TO 3420
CALL
CALL PAKSPL(IFOUR,IFOUR,IASKED(N,2),IPASKED,XFALT)
3420
ITEMP=TEMPA(2)
IF(ITEMP.NE.6)
GO TO 3430
F16 PG.48
IF(TEMPA(3).LT.900.)
GO TO 3430
IF(ASKSPD(N,2).GT.1.)
ASKSPD(N,2)=ASKSPD(N,2)/3600.
TEMPS = ASKSPD(N,2)*3600.
CALL
CALL PAKSPL(ISIX,ISIX,IASKED(N,2),IPASKED,TEMPS)
F16 PG.47
C**** DELETE OLD FLIGHT MODE -- MOVE SCHEDULE UP (NNCT=1)
3430
NTEMP=NNA(N)-IONE
IATRF22(N)=IATRF22(N)-1
DO 3435 K=1,NTEMP
KK=K+IONE
IASKED(N,K)=IASKED(N,KK)
3435
ASKSPD(N,K)=ASKSPD(N,KK)
NNA(N)=NTEMP
IF(NHOLDP(N).GT.0)
TOP OF NEXT PAGE

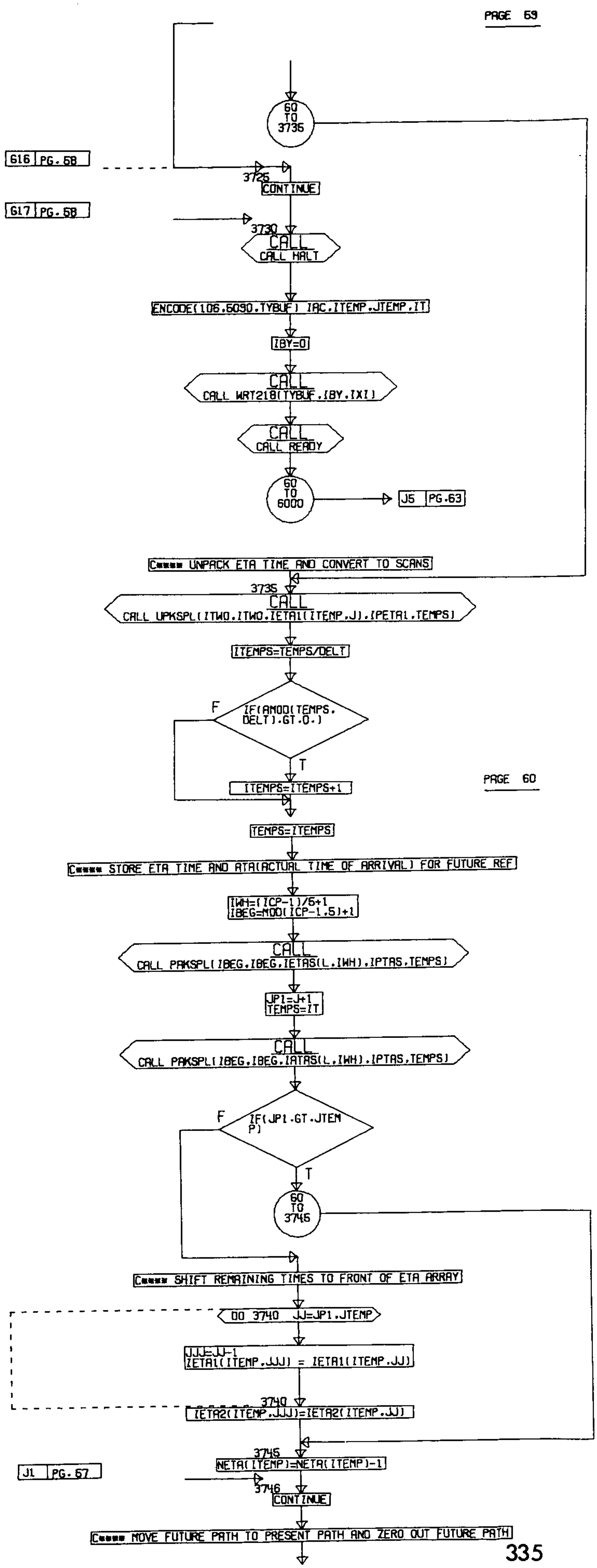

PAGE 49
NHOLDP(N)=NHOLDP(N)-1
BOTTOM OF PRECEDING PAGE
GO TO 2000
C6 PG.20
C**** AIRCRAFT HAS NOT RECEIVED SUFFICIENT CONTROLLER INSTRUCTION -
C**** INSERT CONTINUE UNTIL CLEARANCE RECEIVED CP MODE
PG. 48
3450
TEMPA(1)=8.
TEMPA(2)=0.
TEMPA(3)=0.
CALL
CALL PAKSPL(IFIVE,ISEVEN,IASKED(N,1),IPASKED,TEMPA)
GO TO 2000
C6 PG.20
C**** SPEED AND ALTITUDE
PG. 39 .42.46.48
3500
ALT=ATRAF4(N)
SPD=ATRAF5(N)
SPDF=ASKSPD(N,1)
ALTF=XFALT
C**** IS DESIRED ALTITUDE GREATER THAN RUNWAY HEIGHT
IF(ALTF.GT.RWY Z(IHEN))
F
T
GO TO 3509
G1 PG.51
PAGE 50
C**** IS THIS AN ARRIVAL
IF(IATRF11(N). EQ.2)
F
T
GO TO 3506
C**** SET CHANGE OF ALTITUDE = 0 -- SET CHANGE OF SPEED
C**** TO TAKE OFF ACCELERATION
DELZ=0.
CALL
CALL UPKUNI(ICLASS,ISIX,IPERFS1,1.,DELSP)
CALL
CALL UPKUNI(ICLASS,ITEN,IPERFT,1.,TEMPS)
DELSP=DELSP/3600./TEMPS
GO TO 3600
G8 PG.54
3506
ILOC=NFIX+KILS-2
CALL
CALL UPKSPL(IONE,ITWO,IFIX(ILOC),IPNSITE,TEMPA)
RX=XN-TEMPA(1)
RY=YN-TEMPA(2)
C**** COMPUTE RANGE TO CENTER OF RUNWAY (RANGE) -- COMPUTE
C**** SEC TO TOUCHDOWN (SECTD)
RANGE=SQRT(RX*RX+RY*RY)
SECTD=2.*RANGE/(SPD+SPDF)
C**** CHANGE OF ALT. =(ALT-RUNWAY ALTITUDE/SECTD) -- CHANGE
C**** OF SPEED =(FUTURE SPEED-SPEED)/SECTD


PAGE 51
DELSP=(SPDF-SPD)/SECTD
DELZ=-(ALT-RWYZ(IHEN))/SECTD
GO TO 3600
G8 PG.54
G1 PG.49
C**** HAS DESIRED ALT. BEEN REACHED
3509
IF(ABS(ALT-ALT F).GT.DELALT)
F
T
GO TO 3525
DELZ=0.
GO TO 3550
G5 PG.53
3525
RCLIMB=(ALTF-ALT)/DELT
C**** ALT GT ALTF
IF(ALT.LT.ALTF)
F
T
GO TO 3540
G3 PG.52
PAGE 52
C**** SPDF = DESCEND SPEED -- DELZ = DESCEND RATE
CALL
CALL UPKUNI(ICLASS,IEIGHT,IPERFR3,.01,DELZ)
DELZ=AMAX1(-DELZ/60.,RCLIMB)
CALL
CALL UPKUNI(ICLASS,IEIGHT,IPERFR2,.01,TEMPS)
IF(ALT.GT.1150 0.)
F
T
DELZ=AMAX1(-TEMPS/60.,RCLIMB)
IF(ALT.GT.6500. .AND.IATRAF8(N) .EQ.1)
F
T
DELZ=AMAX1(- TEMPS/60.,RCLIMB)
GO TO 3550
G5 PG.53
G3 PG.51
C**** IS ALT GREATER THAN 10000 FT
3540
IF(ALT.GT.1000 0.)
F
T
GO TO 3545
G4 PG.53

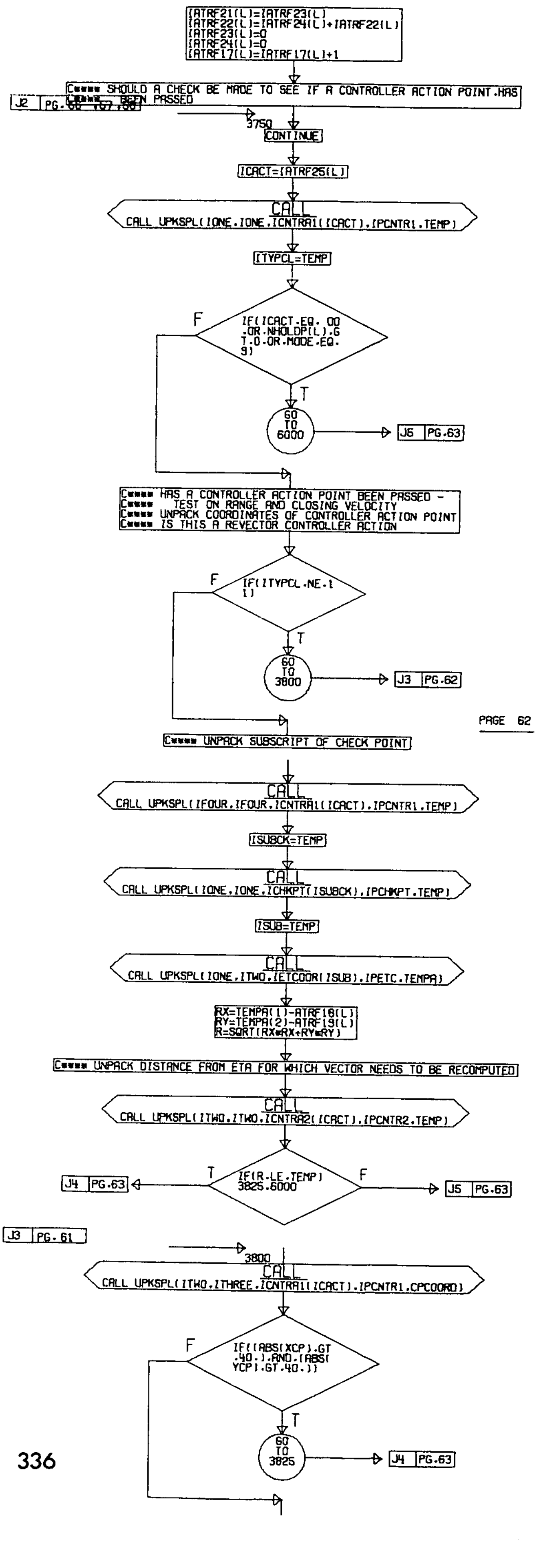

C**** SPDF= CLIMB SPEED BELOW 10000 FT. — DELZ=CLIMB RATE
C**** BELOW 10000 FT.
CALL
CALL UPKUNI(ICLASS.ISIX.IPERFS2.1..SPDF)
SPDF=SPDF/3600.
CALL
CALL UPKUNI(ICLASS.IEIGHT.IPERFR4..01.DELZ)
DELZ=AMIN1(DELZ/60..RCLIMB)
GO TO 3550
G4 PG.52
3545
CALL
CALL UPKUNI(ICLASS.ISIX.IPERFS3.1..SPDF)
C**** SPDF = CLIMBING SPEED ABOVE 10000 — DELZ = CLIMB RATE
C**** ABOVE 10000
SPDF=SPDF/3600.
CALL
CALL UPKUNI(ICLASS.IEIGHT.IPERFR6..01.DELZ)
G5 PG.51 .52
DELZ=AMIN1(DELZ/60..RCLIMB)
3550
ACCRT=(SPDF-SPD)/DELT
C**** FUTURE SPEED GT PRESENT SPEED
IF(ACCRT) 3575 .3575.3590
-
+
0
G6 PG.54
G7 PG.54
PAGE 54
G6 PG.54
G6 PG.53 .53
C**** DELSP=DECELERATION RATE
3575
CALL
CALL UPKUNI(ICLASS.IEIGHT.IPERFR7.1..TEMPS)
TEMPS=-TEMPS/3600./60.
DELSP=AMAX1(ACCRT.TEMPS)
GO TO 3600
G7 PG.53
C**** DELSP=ACCELERATION RATE
3590
CALL
CALL UPKUNI(ICLASS.IEIGHT.IPERFR6.1..TEMPS)
TEMPS=TEMPS/3600.
DELSP=AMIN1(ACCRT.TEMPS/60.)
C**** COMPUTE NEW SPEED SPD=SPD+DELT*DELSP
C**** COMPUTE NEW ALTITUDE ALT=ALT+DELT*DELZ
G8 PG.50 .51
3600
SPD=SPD+DELT*DELSP
ALT=ALT+DELT*DELZ
ALT=AMAX1(ALT.RWYZ(IHEM))
C**** UPDATE ACTIVE TRAFFIC SAMPLE ARRAY
C**** COMPUTE NEW X. Y. VX. VY
C**** UPDATE ACTIVE TRAFFIC ARRAY ON L (L=L+1)
L=L+1
ATRAF(L.1)=ATRAF(N.1)
TEMP=(HEADN+HDOT*DELT/2.)*DTR
VX=SPD*COS(TEMP)
VY=SPD*SIN(TEMP)
ATRAF(L.4)=ALT
ATRAF(L.5)=SPD
ATRAF(L.6)=CHEAD
IATRAF(L.7)=ICLASS
DO 3660 K=8.12
3660
IATRAF(L.K)=IATRAF(N.K)
NHOLOP(L)=NHOLOP(N)
T(L)=T(N)
OLDR(L) = OLDR(N)
XPL(L) = XPL(N)
YPL(L) = YPL(N)
DELH(L)=DELH(N)
HDBEG(L)=HDBEG(N)
NNAT=NNA(N)
NNA(L)=NNAT
DO 3675 I=1.NNAT
IASKED(L.I)=IASKED(N.I)
3675
ASKSPD(L.I)=ASKSPD(N.I)
ATRAF(L.2)=VX*DELT+XN
ATRAF(L.3)=VY*DELT+YN
ATRAF(L.13) =VX
ATRAF(L.14)=VY
ATRAF(L.15) = DELZ
IATRAF(L.16)=MODE*100+MODECP
IATRF17(L)=IATRF17(N)
RX=ATRAF2(L)-RAD(1)
RY=ATRAF3(L)-RAD(2)
RZ=(ATRAF4(L)-RAD(3))/6076.1
RANGE=SQRT(RX*RX+RY*RY+RZ*RZ)
RTHETA=AMOD(ATAN2(RX.RY)*RTD+360..360.)
RTHETA=AMOD(450.-RTHETA.360.)*DTR
DO 3690 I=1.3
Y2=0.
DO 3680 IK=1.12
CALL
CALL RANDU(IARG.IY.TEMP)
IARG=IY
3680
Y2=Y2+TEMP
3690
TEMPA(I)=(Y2-6.)*SIG(I)
ATRF18(L)=XN+TEMPA(1)*COS(RTHETA)-TEMPA(2)*SIN(RTHETA)*RANG
ATRF19(L)=YN+TEMPA(1)*SIN(RTHETA)+TEMPA(2)*COS(RTHETA)*RANG
ATRF18(L)=ATRAF2(L)+ATRF18(L)-XN
ATRF19(L)=ATRAF3(L)+ATRF19(L)-YN
TEMP=50.
IF(TEMPA(3).LT .0.)
F
T
TEMP=-50.
ITEMP=(TEMPA(3)+TEMP)/100.
TEMP=ITEMP*100
ATRF20(L) = ATRAF4(L)+TEMP
DO 3700 I=21.26
3700
IATRAF(L.I)=IATRAF(N.I)
ICP = IATRAF(L.17)
DO 3715 I=1.2
IATAS(L.I)=IATAS(N.I)
3715
IETAS(L.I)=IETAS(N.I)
C**** HAS AN ETA POINT BEEN PASSED
IF(MODE.EQ.9.OR.NHOLOP(L).GT .0)
F
T
GO TO 3750
J2 PG.61
C**** UNPACK ARRAY SUBSCRIPT OF ETA
CALL
CALL UPKSPL(IFIVE.IFIVE.IATRF21(L).IPAPATH.TEMP)
ITEMP=TEMP

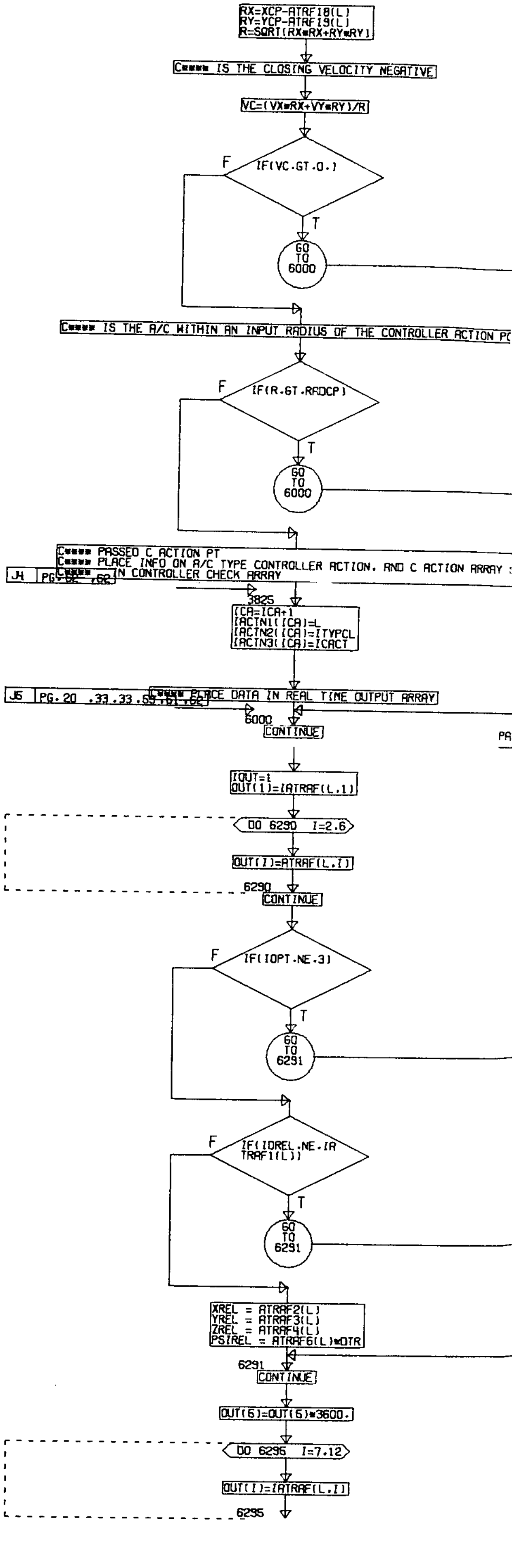


PAGE 57
IF(ITEMP .EQ. 0)
F
T
GO TO 3746
J1 PG.60
IF(MODE.EQ.10.AND.IATRF23(L).NE.0)
GO TO 3720
G15 PG.58
CALL
CALL UPKSPL(IONE.ITWO.IETCOOR(ITEMP).IPETC.TEMPA)
RX=ATRAF2(L)-TEMPA(1)
RY=ATRAF3(L)-TEMPA(2)
R=SQRT(RX*RX+RY*RY)
VC=-(VX*RX+VY*RY)/R
C**** IS THE A/C FLYING AWAY FROM THE ETA
IF(VC.GT.0.)
GO TO 3750
J2 PG.61
C**** IS THE A/C WITHIN AN INPUT RADIUS OF THE ETA POINT
PAGE 58
IF(R.GT.RADCP)
GO TO 3750
J2 PG.61
PG.57
3720
CONTINUE
JTEMP=NETA(ITEMP)
IF(JTEMP.EQ.0)
GO TO 3730
G17 PG.59
C**** LOCATE A/C ID IN ETA ARRAY
DO 3725 J=1.JTEMP
G16 PG.59
CALL UPKSPL(IONE.IONE.IETA1(ITEMP.J).IPETA1.TEMPS)
ITEMPS=TEMPS
IF(ITEMPS.EQ.IAC)

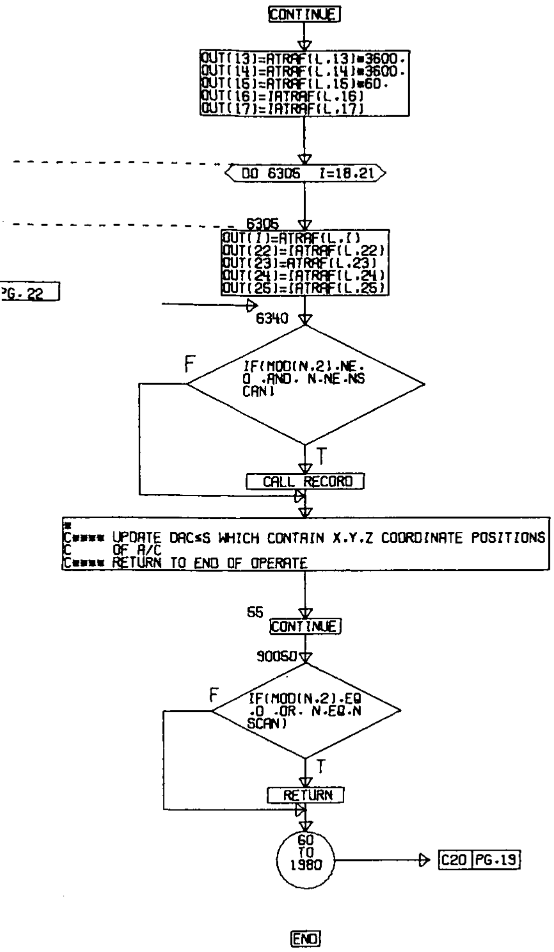


PAGE 59
GO TO 3735
G16 PG.58
3725
CONTINUE
G17 PG.58
3730
CALL HALT
ENCODE(106.5090.TYBUF) IAC.ITEMP.JTEMP.IT
IBY=0
CALL WRT218(TYBUF.IBY.IXI)
CALL READY
GO TO 6000
J5 PG.63
C**** UNPACK ETA TIME AND CONVERT TO SCANS
3735
CALL UPKSPL(ITWO.ITWO.IETA1(ITEMP.J).IPETA1.TEMPS)
ITEMPS=TEMPS/DELT
IF(AMOD(TEMPS.DELT).GT.0.)
ITEMPS=ITEMPS+1
PAGE 60
TEMPS=ITEMPS
C**** STORE ETA TIME AND ATA(ACTUAL TIME OF ARRIVAL) FOR FUTURE REF
IWH=(ICP-1)/5+1
IBEG=MOD(ICP-1.5)+1
CALL PAKSPL(IBEG.IBEG.IETAS(L.IWH).IPTAS.TEMPS)
JP1=J+1
TEMPS=IT
CALL PAKSPL(IBEG.IBEG.IATAS(L.IWH).IPTAS.TEMPS)
IF(JP1.GT.JTEMP)
GO TO 3745
C**** SHIFT REMAINING TIMES TO FRONT OF ETA ARRAY
DO 3740 JJ=JP1.JTEMP
JJJ=JJ-1
IETA1(ITEMP.JJJ) = IETA1(ITEMP.JJ)
3740
IETA2(ITEMP.JJJ)=IETA2(ITEMP.JJ)
3745
NETA(ITEMP)=NETA(ITEMP)-1
J1 PG.57
3746
CONTINUE
C**** MOVE FUTURE PATH TO PRESENT PATH AND ZERO OUT FUTURE PATH

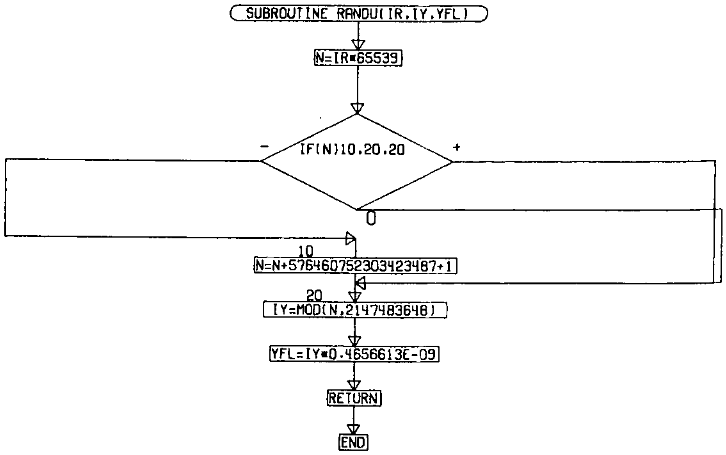

IATRF21(L)=IATRF23(L)
IATRF22(L)=IATRF24(L)+IATRF22(L)
IATRF23(L)=0
IATRF24(L)=0
IATRF17(L)=IATRF17(L)+1
C**** SHOULD A CHECK BE MADE TO SEE IF A CONTROLLER ACTION POINT HAS
C**** BEEN PASSED
J2 PG.60 .67 .68
3750
CONTINUE
ICACT=IATRF25(L)
CALL
CALL UPKSPL(IONE,IONE,ICNTRA1(ICACT),IPCNTR1,TEMP)
ITYPCL=TEMP
F
IF(ICACT.EQ. 00 .OR.NHOLDP(L).G T.0.OR.MODE.EQ. 9)
T
GO TO 6000
J5 PG.63
C**** HAS A CONTROLLER ACTION POINT BEEN PASSED -
C**** TEST ON RANGE AND CLOSING VELOCITY
C**** UNPACK COORDINATES OF CONTROLLER ACTION POINT
C**** IS THIS A REVECTOR CONTROLLER ACTION
F
IF(ITYPCL.NE.1 1)
T
GO TO 3800
J3 PG.62
PAGE 62
C**** UNPACK SUBSCRIPT OF CHECK POINT
CALL
CALL UPKSPL(IFOUR,IFOUR,ICNTRA1(ICACT),IPCNTR1,TEMP)
ISUBCK=TEMP
CALL
CALL UPKSPL(IONE,IONE,ICHKPT(ISUBCK),IPCHKPT,TEMP)
ISUB=TEMP
CALL
CALL UPKSPL(IONE,ITWO,IETCOOR(ISUB),IPETC,TEMPA)
RX=TEMPA(1)-ATRF18(L)
RY=TEMPA(2)-ATRF19(L)
R=SQRT(RX*RX+RY*RY)
C**** UNPACK DISTANCE FROM ETA FOR WHICH VECTOR NEEDS TO BE RECOMPUTED
CALL
CALL UPKSPL(ITWO,ITWO,ICNTRA2(ICACT),IPCNTR2,TEMP)
J4 PG.63
T
IF(R.LE.TEMP) 3825,6000
F
J5 PG.63
J3 PG.61
3800
CALL
CALL UPKSPL(ITWO,ITHREE,ICNTRA1(ICACT),IPCNTR1,CPCOORD)
F
IF((ABS(XCP).GT .40.).AND.(ABS(YCP).GT.40.))
T
GO TO 3825
J4 PG.63


RX=XCP-ATRF18(L)
RY=YCP-ATRF19(L)
R=SQRT(RX*RX+RY*RY)
C**** IS THE CLOSING VELOCITY NEGATIVE
VC=(VX*RX+VY*RY)/R
F
IF(VC.GT.0.)
T
GO TO 6000
C**** IS THE A/C WITHIN AN INPUT RADIUS OF THE CONTROLLER ACTION PO
F
IF(R.GT.RADCP)
T
GO TO 6000
C**** PASSED C ACTION PT
C**** PLACE INFO ON A/C TYPE CONTROLLER ACTION, AND C ACTION ARRAY
C**** IN CONTROLLER CHECK ARRAY
J4 PG.62 .62
3825
ICA=ICA+1
IACTN1(ICA)=L
IACTN2(ICA)=ITYPCL
IACTN3(ICA)=ICACT
J5 PG.20 .33 .33 .59 .61 .62
C**** PLACE DATA IN REAL TIME OUTPUT ARRAY
6000
CONTINUE
IOUT=1
OUT(1)=IATRAF(L,1)
DO 6290 I=2,6
OUT(I)=ATRAF(L,I)
6290
CONTINUE
F
IF(IOPT.NE.3)
T
GO TO 6291
F
IF(IOREL.NE.IA TRAF1(L))
T
GO TO 6291
XREL = ATRAF2(L)
YREL = ATRAF3(L)
ZREL = ATRAF4(L)
PSIREL = ATRAF6(L)*DTR
6291
CONTINUE
OUT(6)=OUT(6)*3600.
DO 6295 I=7,12
OUT(I)=IATRAF(L,I)
6295

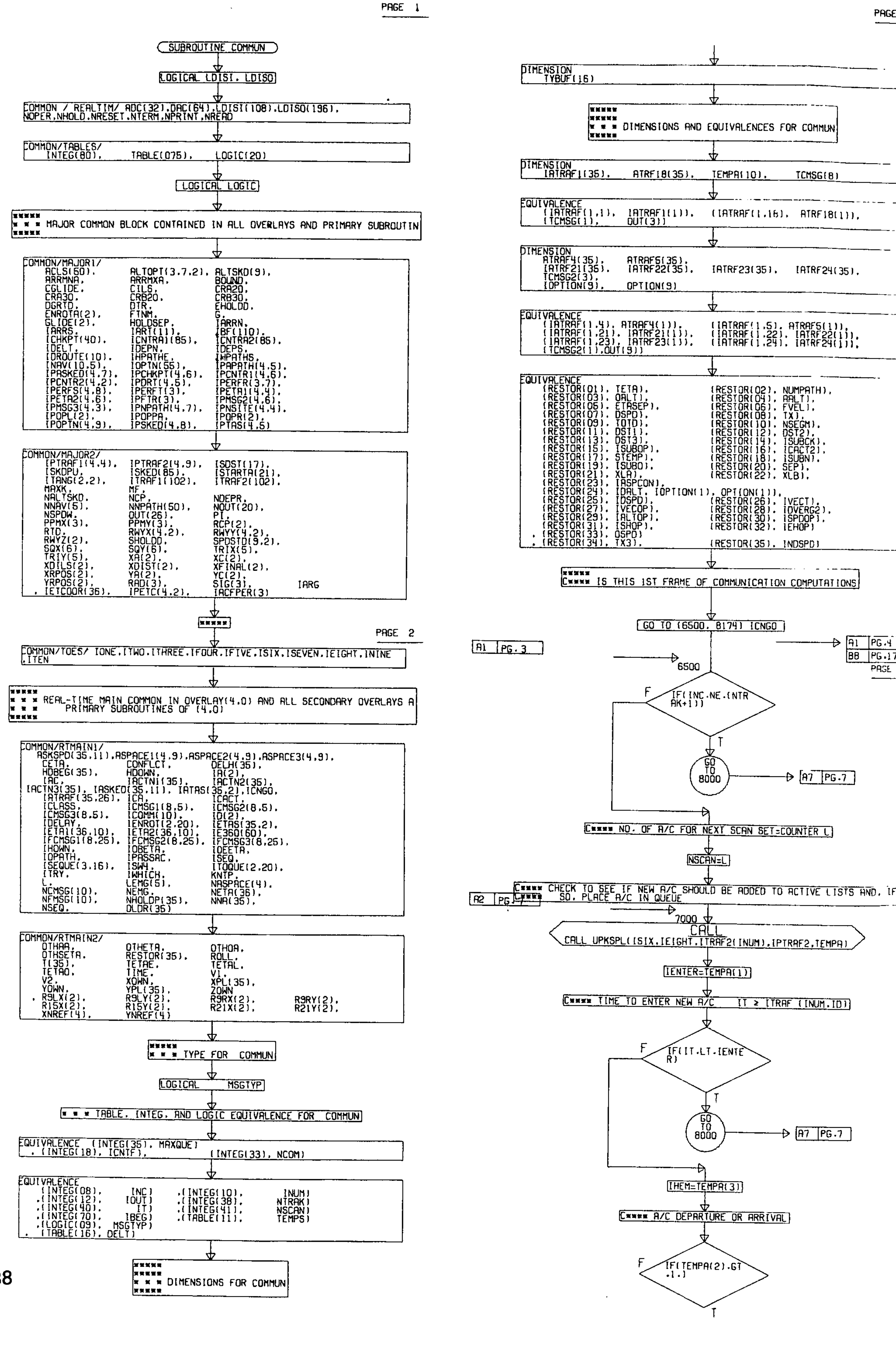
PAGE 65

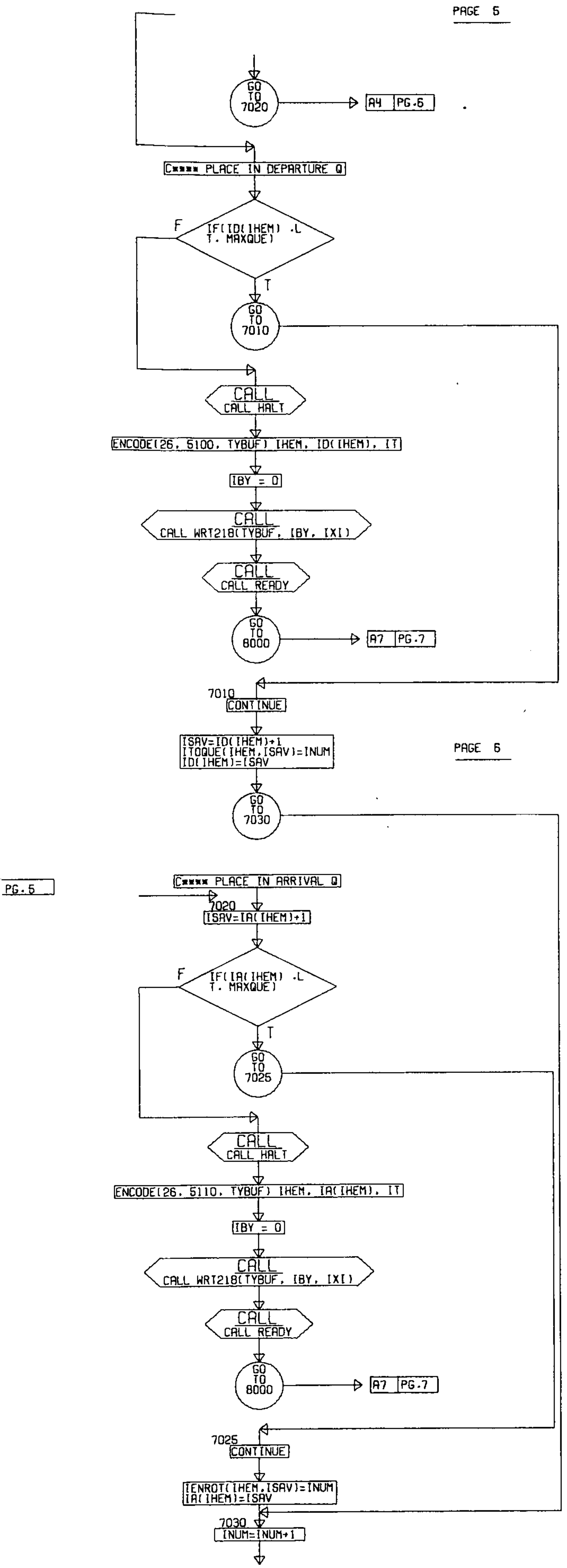


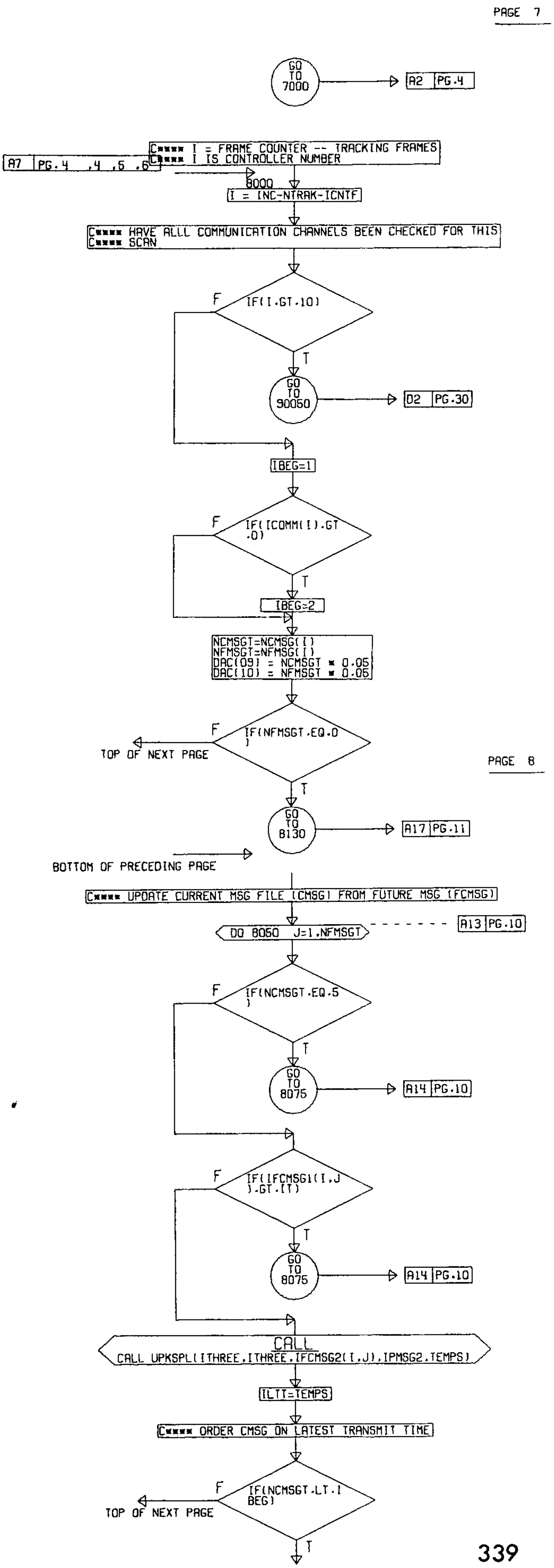
PAGE 1

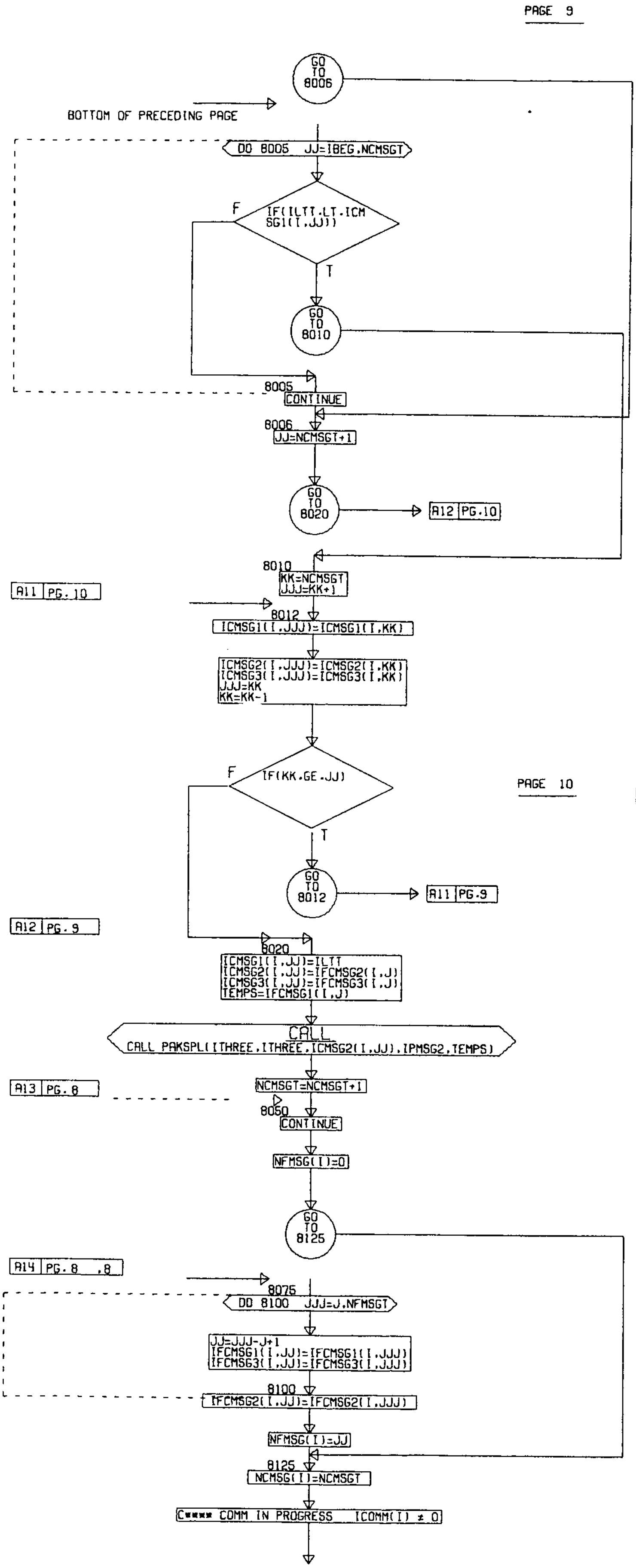


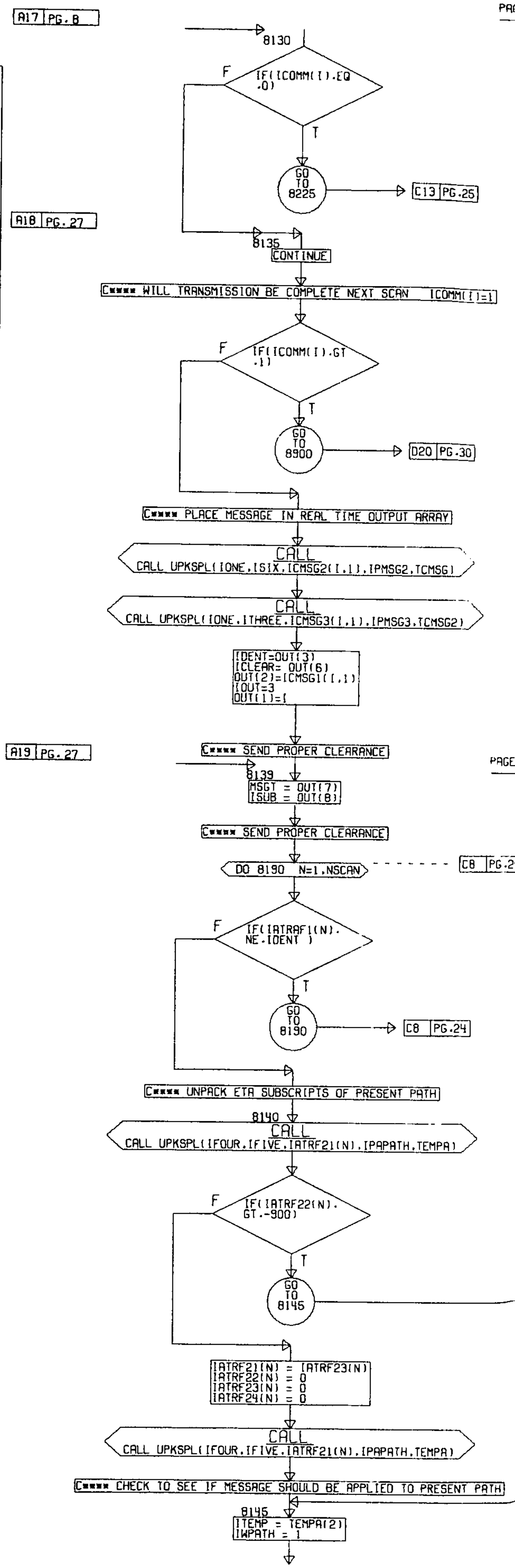
CROSSHAIR NO. 1


PAGE 1
SUBROUTINE COMMUN
LOGICAL LDISI, LDISO
COMMON / REALTIM/ ADC(32),DAC(64),LDISI(108),LDISO(196), NOPER,NHOLD,NRESET,NTERM,NPRINT,NREAD
COMMON/TABLES/ INTEG(80), TABLE(075), LOGIC(20)
LOGICAL LOGIC
***** * * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN *****
COMMON/MAJOR1/ ACLS(50), ALTOPT(3,7,2), ALTSKD(9), ARRMNA, ARRMXA, BOUND, CGLIDE, CILS, CRA20, CRA30, CRB20, CRB30, DGRTD, DTR, EHOLDD, ENROTA(2), FTNM, G, GLIDE(2), HOLDSEP, IARRN, IARRS, IART(11), IBF(110), ICHKPT(40), ICNTRA1(85), ICNTRA2(85), IDELT, IDEPN, IDEPS, IDROUTE(10), IHPATHE, IHPATHS, INAV(10,5), IOPTN(55), IPAPATH(4,5), IPASKED(4,7), IPCHKPT(4,6), IPCNTR1(4,6), IPCNTR2(4,2), IPDRT(4,5), IPERFR(3,7), IPERFS(4,8), IPERFT(3), IPETA1(4,4), IPETA2(4,6), IPFTR(3), IPMSG2(4,6), IPMSG3(4,3), IPNPATH(4,7), IPNSITE(4,4), IPOPL(2), IPOPPA, IPOPR(2), IPOPTN(4,9), IPSKED(4,8), IPTAS(4,5)
COMMON/MAJOR2/ IPTRAF1(4,4), IPTRAF2(4,9), ISDST(17), ISKDPU, ISKED(85), ISTARTA(21), ITANG(2,2), ITRAF1(102), ITRAF2(102), MAXK, MF, NALTSKD, NCP, NDEPR, NNAV(5), NNPATH(50), NOUT(20), NSPDW, OUT(26), PI, PPMX(3), PPMY(3), RCP(2), RTD, RWYX(4,2), RWYY(4,2), RWYZ(2), SHOLDD, SPDSTD(9,2), SQX(6), SQY(6), TRIX(5), TRIY(5), XA(2), XC(2), XDILS(2), XDIST(2), XFINAL(2), XRPOS(2), YA(2), YC(2), YRPOS(2), RAD(3), SIG(3), IARG . IETCOOR(36), IPETC(4,2), IACFPER(3)
*****
PAGE 2
COMMON/TOES/ IONE,ITWO,ITHREE,IFOUR,IFIVE,ISIX,ISEVEN,IEIGHT,ININE .ITEN
***** * * * REAL-TIME MAIN COMMON IN OVERLAY(4,0) AND ALL SECONDARY OVERLAYS A * * * PRIMARY SUBROUTINES OF (4,0) *****
COMMON/RTMAIN1/ ASKSPD(35,11),ASPACE1(4,9),ASPACE2(4,9),ASPACE3(4,9), CETA, CONFLCT, DELH(35), HDBEG(35), HDOWN, IA(2), IAC, IACTN1(35), IACTN2(35), IACTN3(35), IASKED(35,11), IATRS(35,2),ICNGO, IATRAF(35,26), ICA, ICACT, ICLASS, ICMSG1(8,5), ICMSG2(8,5), ICMSG3(8,5), ICOMM(10), ID(2), IDELAY, IENROT(2,20), IETAS(35,2), IETA1(36,10), IETA2(36,10), IE360(60), IFCMSG1(8,25), IFCMSG2(8,25), IFCMSG3(8,25), IHOWN, IOBETA, IOEETA, IOPATH, IPASSAC, ISEQ, ISEQUE(3,16), ISW4, ITOQUE(2,20), ITRY, IWHICH, KNTP, L, LEMG(5), NASPACE(4), NCMSG(10), NEMG, NETA(36), NFMSG(10), NHOLDP(35), NNA(35), NSEQ, OLDR(35)
COMMON/RTMAIN2/ OTHAA, OTHETA, OTHOA, OTHSETA, RESTOR(35), ROLL, T(35), TETAE, TETAL, TETAO, TIME, VI, V2, XOWN, XPL(35), YOWN, YPL(35), ZOWN . R9LX(2), R9LY(2), R9RX(2), R9RY(2), R15X(2), R15Y(2), R21X(2), R21Y(2), XNREF(4), YNREF(4)
***** * * * TYPE FOR COMMUN
LOGICAL MSGTYP
* * * TABLE, INTEG, AND LOGIC EQUIVALENCE FOR COMMUN
EQUIVALENCE (INTEG(35), MAXQUE) . (INTEG(18), ICNTF), (INTEG(33), NCOM)
EQUIVALENCE (INTEG(08), INC) .(INTEG(10), INUM) .(INTEG(12), IOUT) .(INTEG(38), NTRAK) .(INTEG(40), IT) .(INTEG(41), NSCAN) .(INTEG(70), IBEG) .(TABLE(11), TEMPS) .(LOGIC(09), MSGTYP) . (TABLE(16), DELT)
***** ***** * * * DIMENSIONS FOR COMMUN *****
PAGE
DIMENSION TYBUF(16)
***** ***** * * * DIMENSIONS AND EQUIVALENCES FOR COMMUN *****
DIMENSION IATRAF1(35), ATRF18(35), TEMPA(10), TCMSG(8)
EQUIVALENCE (IATRAF(1,1), IATRAF1(1)), (IATRAF(1,16), ATRF18(1)), (TCMSG(1), OUT(3))
DIMENSION ATRAF4(35), ATRAF5(35), IATRF21(35), IATRF22(35), IATRF23(35), IATRF24(35), TCMSG2(3), IOPTION(9), OPTION(9)
EQUIVALENCE (IATRAF(1,4), ATRAF4(1)), (IATRAF(1,5), ATRAF5(1)), (IATRAF(1,21), IATRF21(1)), (IATRAF(1,22), IATRF22(1)), (IATRAF(1,23), IATRF23(1)), (IATRAF(1,24), IATRF24(1)), (TCMSG2(1),OUT(9))
EQUIVALENCE (RESTOR(01), TETA), (RESTOR(02), NUMPATH), (RESTOR(03), OALT), (RESTOR(04), AALT), (RESTOR(05), ETASEP), (RESTOR(06), FVEL), (RESTOR(07), DSPD), (RESTOR(08), TX), (RESTOR(09), TOTD), (RESTOR(10), NSEGM), (RESTOR(11), DST1), (RESTOR(12), DST2), (RESTOR(13), DST3), (RESTOR(14), ISUBCK), (RESTOR(15), ISUBOP), (RESTOR(16), ICACT2), (RESTOR(17), STEMP), (RESTOR(18), ISUBN), (RESTOR(19), ISUBO), (RESTOR(20), SEP), (RESTOR(21), XLA), (RESTOR(22), XLB), (RESTOR(23), IASPCON), (RESTOR(24), IDALT, IOPTION(1), OPTION(1)), (RESTOR(25), IDSPD), (RESTOR(26), IVECT), (RESTOR(27), IVECOP), (RESTOR(28), IOVERG2), (RESTOR(29), IALTOP), (RESTOR(30), ISPDOP), (RESTOR(31), ISHOP), (RESTOR(32), IEHOP) . (RESTOR(33), OSPD) . (RESTOR(34), TX3), (RESTOR(35), INDSPD)
***** C**** IS THIS 1ST FRAME OF COMMUNICATION COMPUTATIONS
GO TO (6500, 8174) ICNGO
A1 PG.4
B8 PG.17
PAGE
A1 PG.3
6500
IF(INC.NE.(NTRAK+1))
F
T
GO TO 8000
A7 PG.7
C**** NO. OF A/C FOR NEXT SCAN SET=COUNTER L
NSCAN=L
C**** CHECK TO SEE IF NEW A/C SHOULD BE ADDED TO ACTIVE LISTS AND, IF C**** SO, PLACE A/C IN QUEUE
A2 PG
7000
CALL
CALL UPKSPL(ISIX,IEIGHT,ITRAF2(INUM),IPTRAF2,TEMPA)
IENTER=TEMPA(1)
C**** TIME TO ENTER NEW A/C IT ≥ ITRAF (INUM,ID)
IF(IT.LT.IENTER)
F
T
GO TO 8000
A7 PG.7
IHEM=TEMPA(3)
C**** A/C DEPARTURE OR ARRIVAL
IF(TEMPA(2).GT .1.)
F
T

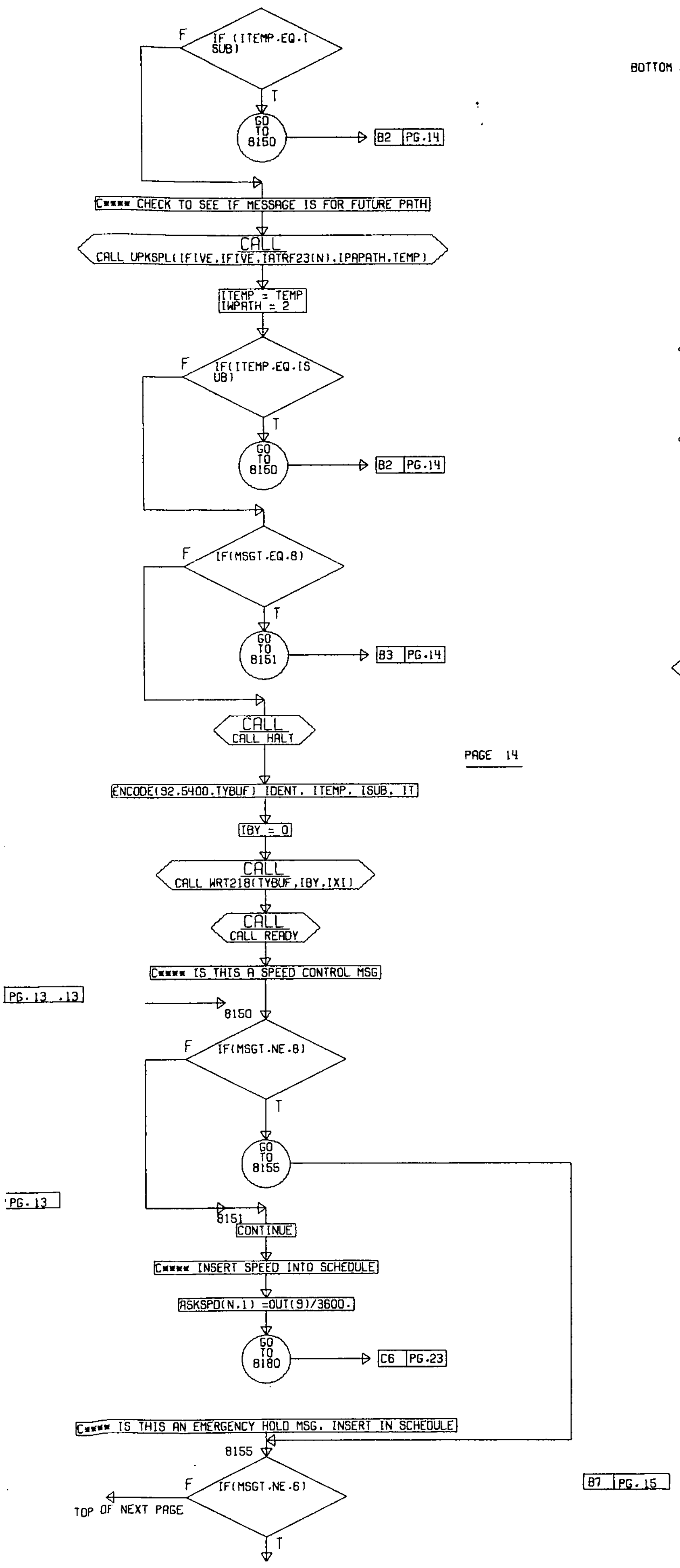

PAGE 5
GO TO 7020
A4 PG.6
C**** PLACE IN DEPARTURE Q
IF(ID(IHEM) .LT. MAXQUE)
F
T
GO TO 7010
CALL
CALL HALT
ENCODE(26, 5100, TYBUF) IHEM, ID(IHEM), IT
IBY = 0
CALL
CALL WRT218(TYBUF, IBY, IXI)
CALL
CALL READY
GO TO 8000
A7 PG.7
7010
CONTINUE
ISAV=ID(IHEM)+1
ITOQUE(IHEM,ISAV)=INUM
ID(IHEM)=ISAV
GO TO 7030
PAGE 6
PG.5
C**** PLACE IN ARRIVAL Q
7020
ISAV=IA(IHEM)+1
IF(IA(IHEM) .LT. MAXQUE)
F
T
GO TO 7025
CALL
CALL HALT
ENCODE(26, 5110, TYBUF) IHEM, IA(IHEM), IT
IBY = 0
CALL
CALL WRT218(TYBUF, IBY, IXI)
CALL
CALL READY
GO TO 8000
A7 PG.7
7025
CONTINUE
IENROT(IHEM,ISAV)=INUM
IA(IHEM)=ISAV
7030
INUM=INUM+1


PAGE 7
GO TO 7000
A2 PG.4
C**** I = FRAME COUNTER -- TRACKING FRAMES
C**** I IS CONTROLLER NUMBER
A7 PG.4 ,4 ,5 ,6
8000
I = INC-NTRAK-ICNTF
C**** HAVE ALLL COMMUNICATION CHANNELS BEEN CHECKED FOR THIS
C**** SCAN
IF(I.GT.10)
F
T
GO TO 90050
D2 PG.30
IBEG=1
IF(ICOMM(I).GT.0)
F
T
IBEG=2
NCMSGT=NCMSG(I)
NFMSGT=NFMSG(I)
DAC(09) = NCMSGT * 0.05
DAC(10) = NFMSGT * 0.05
IF(NFMSGT.EQ.0)
F
TOP OF NEXT PAGE
T
GO TO 8130
A17 PG.11
PAGE 8
BOTTOM OF PRECEDING PAGE
C**** UPDATE CURRENT MSG FILE (CMSG) FROM FUTURE MSG (FCMSG)
DO 8050 J=1,NFMSGT
A13 PG.10
IF(NCMSGT.EQ.5)
F
T
GO TO 8075
A14 PG.10
IF(IFCMSG1(I,J).GT.IT)
F
T
GO TO 8075
A14 PG.10
CALL
CALL UPKSPL(ITHREE,ITHREE,IFCMSG2(I,J),IPMSG2,TEMPS)
ILTT=TEMPS
C**** ORDER CMSG ON LATEST TRANSMIT TIME
IF(NCMSGT.LT.IBEG)
F
TOP OF NEXT PAGE
T

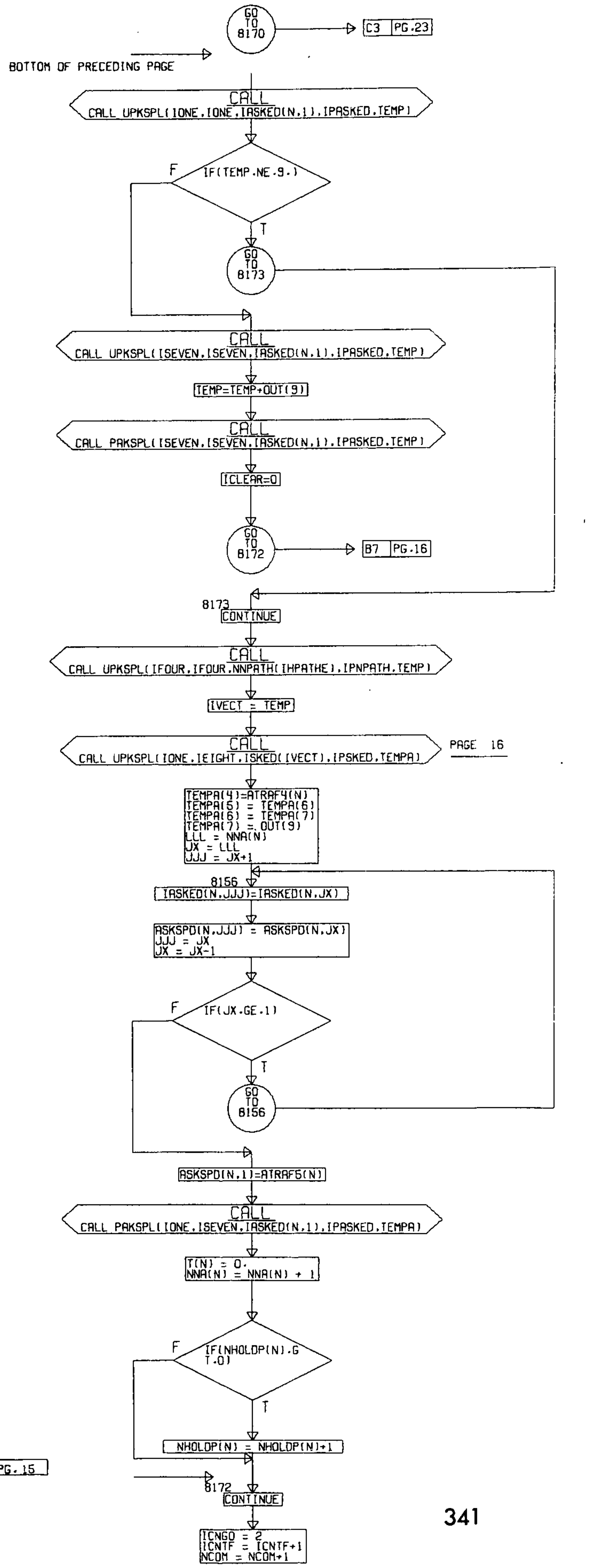

PAGE 9
GO TO 8006
BOTTOM OF PRECEDING PAGE
DO 8005 JJ=IBEG,NCMSGT
IF(ILTT.LT.ICMSG1(I,JJ))
F
T
GO TO 8010
8005 CONTINUE
8006 JJ=NCMSGT+1
GO TO 8020
A12 PG.10
8010 KK=NCMSGT
JJJ=KK+1
A11 PG. 10
8012 ICMSG1(I,JJJ)=ICMSG1(I,KK)
ICMSG2(I,JJJ)=ICMSG2(I,KK)
ICMSG3(I,JJJ)=ICMSG3(I,KK)
JJJ=KK
KK=KK-1
IF(KK.GE.JJ)
PAGE 10
GO TO 8012
A11 PG.9
A12 PG. 9
8020 ICMSG1(I,JJ)=ILTT
ICMSG2(I,JJ)=IFCMSG2(I,J)
ICMSG3(I,JJ)=IFCMSG3(I,J)
TEMPS=IFCMSG1(I,J)
CALL
CALL PAKSPL(ITHREE,ITHREE,ICMSG2(I,JJ),IPMSG2,TEMPS)
A13 PG. 8
NCMSGT=NCMSGT+1
8050 CONTINUE
NFMSG(I)=0
GO TO 8125
A14 PG. 8 ,8
8075
DO 8100 JJJ=J,NFMSGT
JJ=JJJ-J+1
IFCMSG1(I,JJ)=IFCMSG1(I,JJJ)
IFCMSG3(I,JJ)=IFCMSG3(I,JJJ)
8100 IFCMSG2(I,JJ)=IFCMSG2(I,JJJ)
NFMSG(I)=JJ
8125 NCMSG(I)=NCMSGT
C**** COMM IN PROGRESS ICOMM(I) ≠ 0

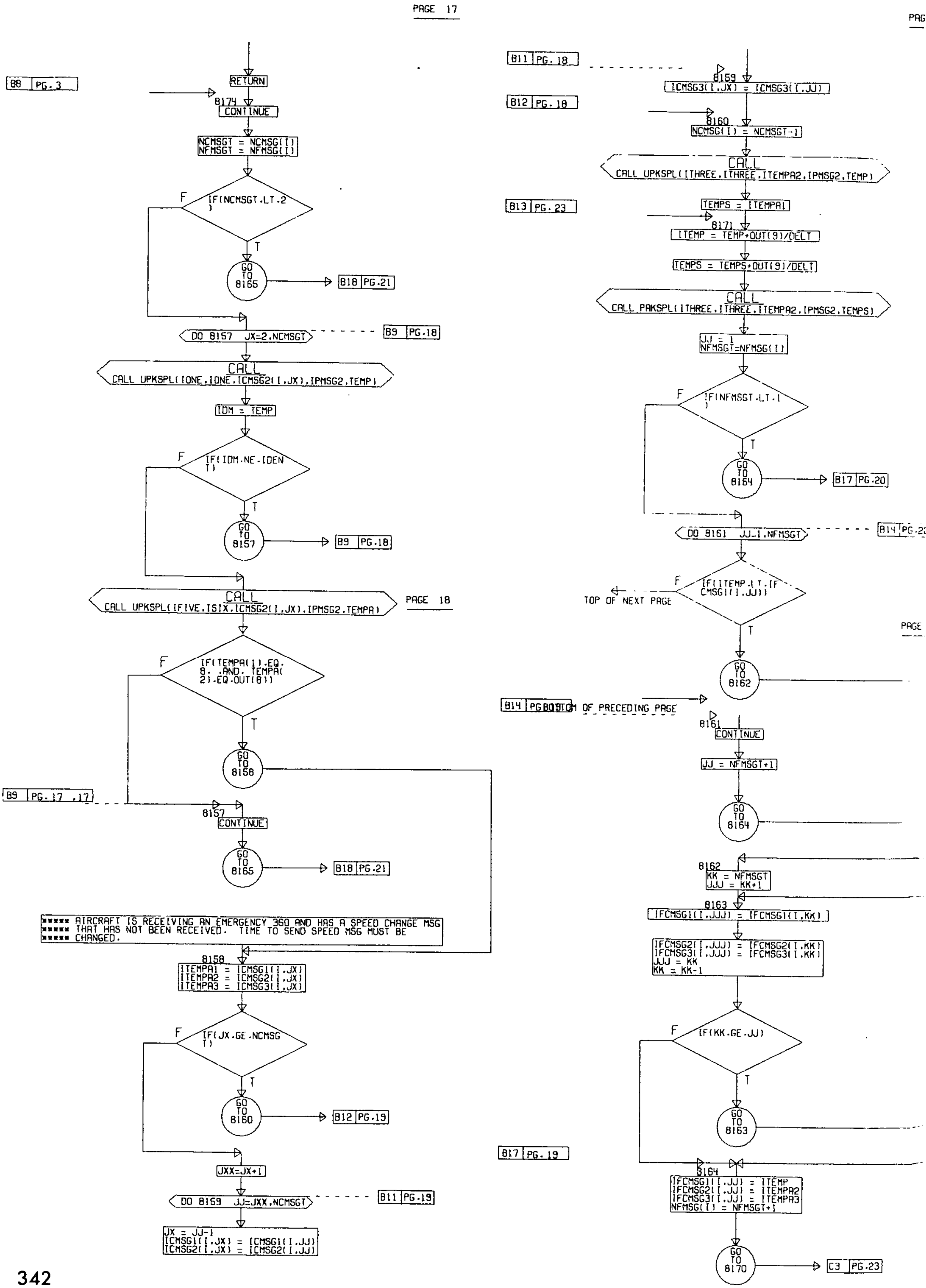

A17 PG. 8
8130
IF(ICOMM(I).EQ.0)
F
T
GO TO 8225
C13 PG.25
A18 PG. 27
8135 CONTINUE
C**** WILL TRANSMISSION BE COMPLETE NEXT SCAN ICOMM(I)=1
IF(ICOMM(I).GT.1)
GO TO 8900
D20 PG.30
C**** PLACE MESSAGE IN REAL TIME OUTPUT ARRAY
CALL
CALL UPKSPL(IONE,ISIX,ICMSG2(I,1),IPMSG2,TCMSG)
CALL
CALL UPKSPL(IONE,ITHREE,ICMSG3(I,1),IPMSG3,TCMSG2)
IDENT=OUT(3)
ICLEAR= OUT(6)
OUT(2)=ICMSG1(I,1)
IOUT=3
OUT(1)=I
C**** SEND PROPER CLEARANCE
A19 PG. 27
8139 MSGT = OUT(7)
ISUB = OUT(8)
C**** SEND PROPER CLEARANCE
DO 8190 N=1,NSCAN
C8 PG.24
IF(IATRAF1(N).NE.IDENT)
GO TO 8190
C8 PG.24
C**** UNPACK ETA SUBSCRIPTS OF PRESENT PATH
8140
CALL
CALL UPKSPL(IFOUR,IFIVE,IATRF21(N),IPAPATH,TEMPA)
IF(IATRF22(N).GT.-900)
GO TO 8145
IATRF21(N) = IATRF23(N)
IATRF22(N) = 0
IATRF23(N) = 0
IATRF24(N) = 0
CALL
CALL UPKSPL(IFOUR,IFIVE,IATRF21(N),IPAPATH,TEMPA)
C**** CHECK TO SEE IF MESSAGE SHOULD BE APPLIED TO PRESENT PATH
8145 ITEMP = TEMPA(2)
IWPATH = 1

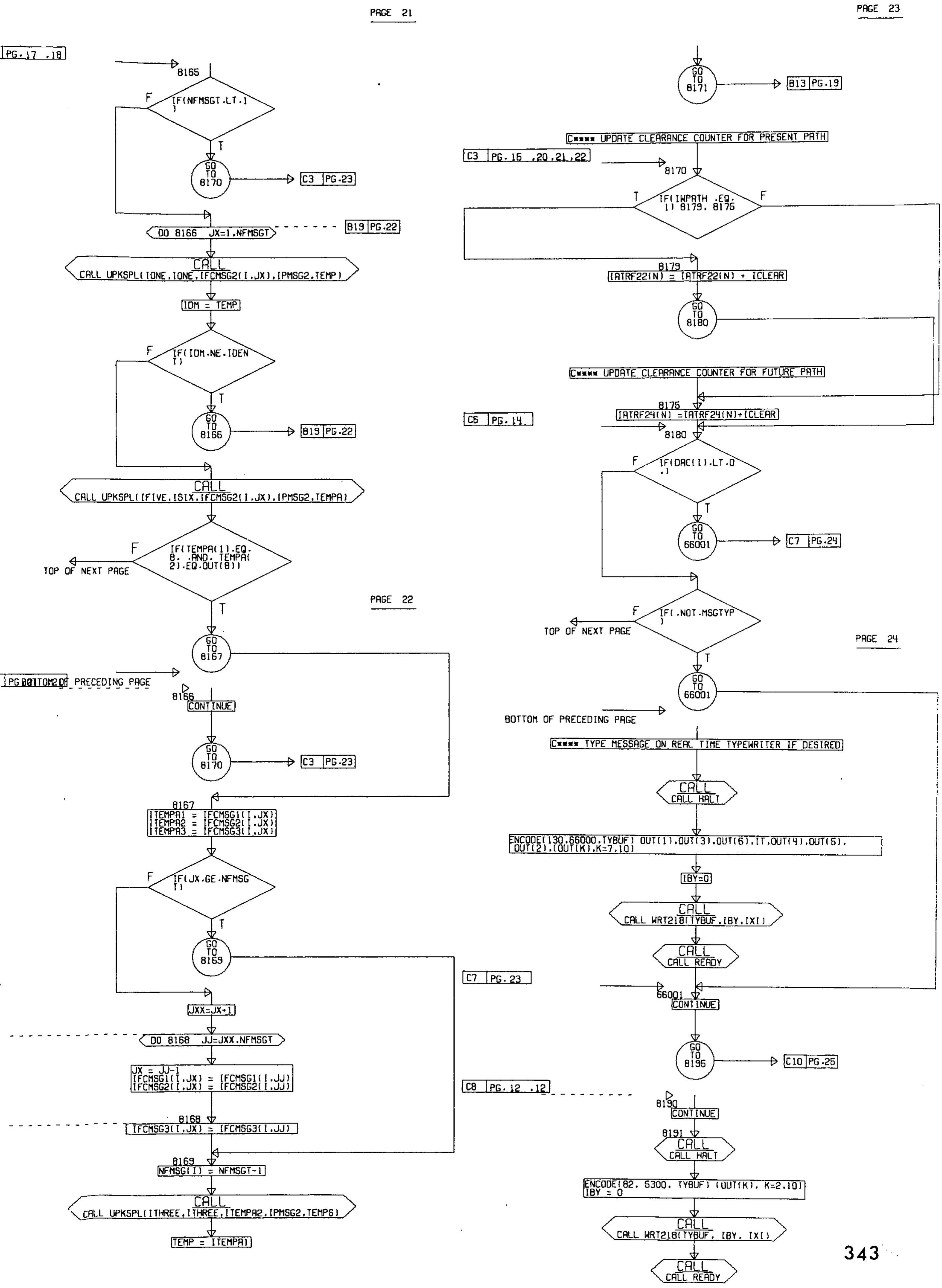
PAGE 13

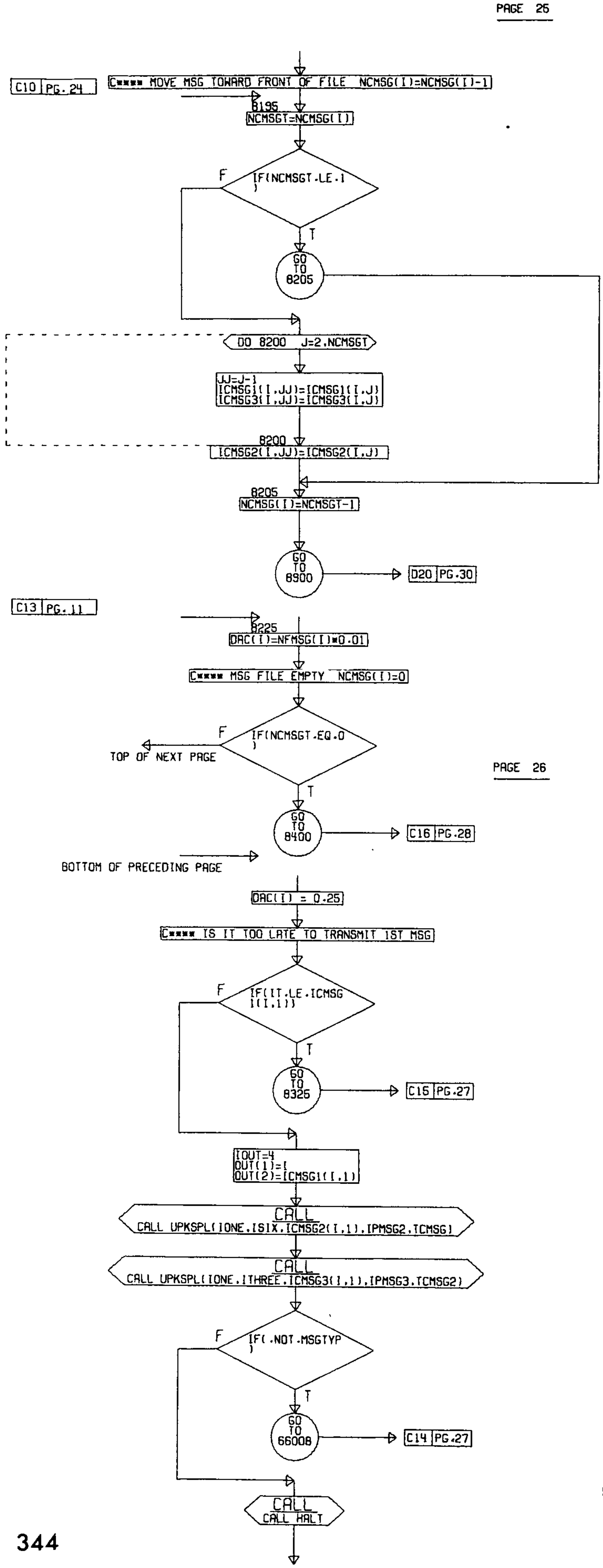


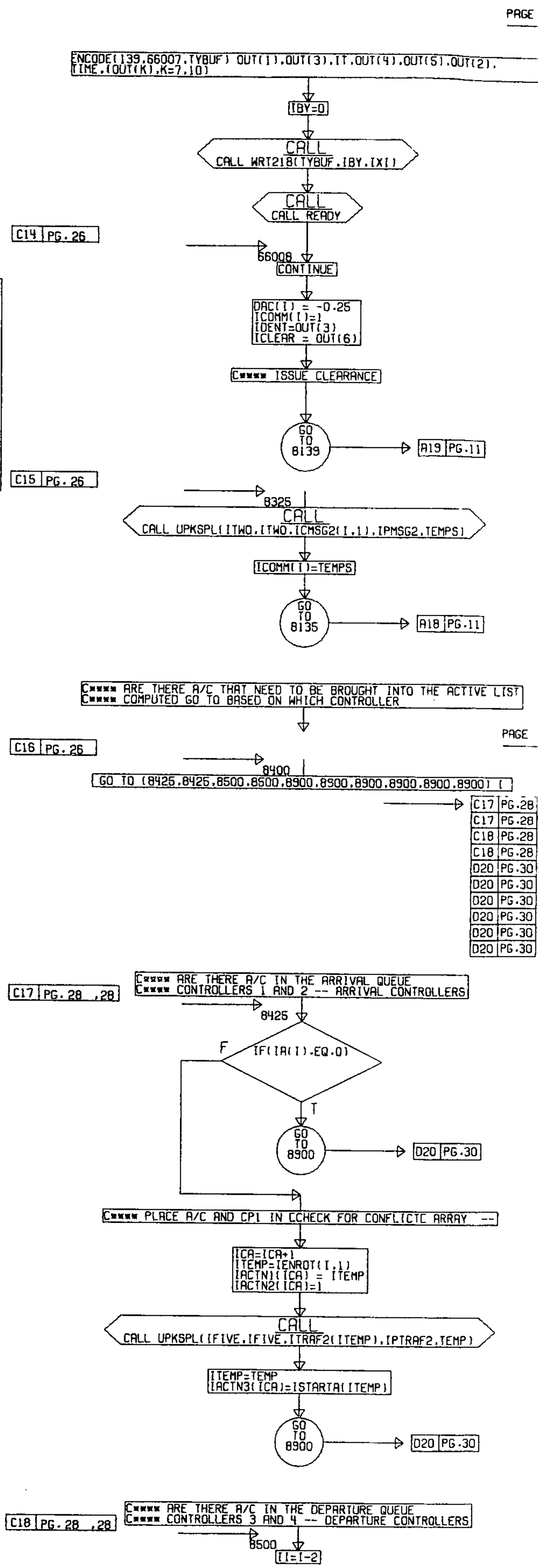
PAGE 15

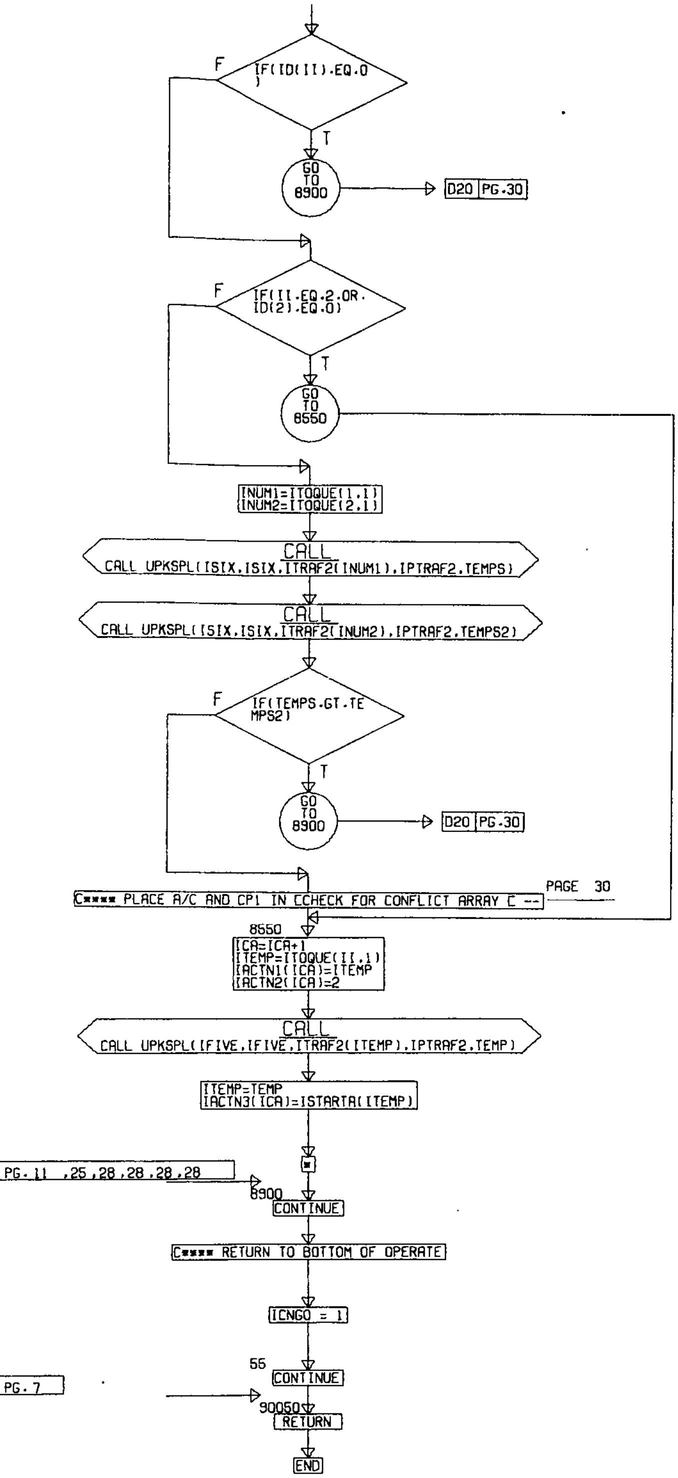



PAGE 17
B8 PG. 3
RETURN
8174
CONTINUE
NCMSGT = NCMSG(I)
NFMSGT = NFMSG(I)
F
IF(NCMSGT.LT.2)
T
GO TO 8165
B18 PG.21
DO 8157 JX=2,NCMSGT
B9 PG.18
CALL
CALL UPKSPL(IONE,IONE,ICMSG2(I,JX),IPMSG2,TEMP)
IDM = TEMP
IF(IDM.NE.IDENT)
GO TO 8157
B9 PG.18
CALL
CALL UPKSPL(IFIVE,ISIX,ICMSG2(I,JX),IPMSG2,TEMPA)
PAGE 18
IF(TEMPA(1).EQ.8. .AND. TEMPA(2).EQ.OUT(8))
GO TO 8158
B9 PG.17 ,17
8157
CONTINUE
GO TO 8165
B18 PG.21
***** AIRCRAFT IS RECEIVING AN EMERGENCY 360 AND HAS A SPEED CHANGE MSG
***** THAT HAS NOT BEEN RECEIVED. TIME TO SEND SPEED MSG MUST BE
***** CHANGED.
8158
ITEMPA1 = ICMSG1(I,JX)
ITEMPA2 = ICMSG2(I,JX)
ITEMPA3 = ICMSG3(I,JX)
IF(JX.GE.NCMSGT)
GO TO 8160
B12 PG.19
JXX=JX+1
DO 8159 JJ=JXX,NCMSGT
B11 PG.19
JX = JJ-1
ICMSG1(I,JX) = ICMSG1(I,JJ)
ICMSG2(I,JX) = ICMSG2(I,JJ)
B11 PG. 18
8159
ICMSG3(I,JX) = ICMSG3(I,JJ)
B12 PG. 18
8160
NCMSG(I) = NCMSGT-1
CALL
CALL UPKSPL(ITHREE,ITHREE,ITEMPA2,IPMSG2,TEMP)
B13 PG. 23
TEMPS = ITEMPA1
8171
ITEMP = TEMP+OUT(9)/DELT
TEMPS = TEMPS+OUT(9)/DELT
CALL
CALL PAKSPL(ITHREE,ITHREE,ITEMPA2,IPMSG2,TEMPS)
JJ = 1
NFMSGT=NFMSG(I)
IF(NFMSGT.LT.1)
GO TO 8164
B17 PG.20
DO 8161 JJ=1,NFMSGT
B14 PG.2
IF(ITEMP.LT.IFCMSG1(I,JJ))
TOP OF NEXT PAGE
GO TO 8162
B14 PG BOTTOM OF PRECEDING PAGE
8161
CONTINUE
JJ = NFMSGT+1
GO TO 8164
8162
KK = NFMSGT
JJJ = KK+1
8163
IFCMSG1(I,JJJ) = IFCMSG1(I,KK)
IFCMSG2(I,JJJ) = IFCMSG2(I,KK)
IFCMSG3(I,JJJ) = IFCMSG3(I,KK)
JJJ = KK
KK = KK-1
IF(KK.GE.JJ)
GO TO 8163
B17 PG. 19
8164
IFCMSG1(I,JJ) = ITEMP
IFCMSG2(I,JJ) = ITEMPA2
IFCMSG3(I,JJ) = ITEMPA3
NFMSG(I) = NFMSGT+1
GO TO 8170
C3 PG.23


PAGE 21
PG.17 .18
8165
IF(NFMSGT.LT.1)
F
T
GO TO 8170
C3 PG.23
DO 8166 JX=1,NFMSGT
B19 PG.22
CALL
CALL UPKSPL(IONE,IONE,IFCMSG2(I,JX),IPMSG2,TEMP)
IDM = TEMP
IF(IDM.NE.IDENT)
F
T
GO TO 8166
B19 PG.22
CALL
CALL UPKSPL(IFIVE,ISIX,IFCMSG2(I,JX),IPMSG2,TEMPA)
IF(TEMPA(1).EQ.8. .AND. TEMPA(2).EQ.OUT(8))
F
TOP OF NEXT PAGE
T
GO TO 8167
PAGE 22
BOTTOM OF PRECEDING PAGE
8166
CONTINUE
GO TO 8170
C3 PG.23
8167
ITEMPA1 = IFCMSG1(I,JX)
ITEMPA2 = IFCMSG2(I,JX)
ITEMPA3 = IFCMSG3(I,JX)
IF(JX.GE.NFMSGT)
F
T
GO TO 8169
JXX=JX+1
DO 8168 JJ=JXX,NFMSGT
JX = JJ-1
IFCMSG1(I,JX) = IFCMSG1(I,JJ)
IFCMSG2(I,JX) = IFCMSG2(I,JJ)
8168
IFCMSG3(I,JX) = IFCMSG3(I,JJ)
8169
NFMSG(I) = NFMSGT-1
CALL
CALL UPKSPL(ITHREE,ITHREE,ITEMPA2,IPMSG2,TEMP6)
TEMP = ITEMPA1
PAGE 23
GO TO 8171
B13 PG.19
C**** UPDATE CLEARANCE COUNTER FOR PRESENT PATH
C3 PG.15 .20 .21 .22
8170
IF(IWPATH .EQ. 1) 8179, 8175
T
F
8179
IATRF22(N) = IATRF22(N) + ICLEAR
GO TO 8180
C**** UPDATE CLEARANCE COUNTER FOR FUTURE PATH
8175
IATRF24(N) =IATRF24(N)+ICLEAR
C6 PG.14
8180
IF(DAC(I).LT.0.)
F
T
GO TO 66001
C7 PG.24
IF(.NOT.MSGTYP)
F
TOP OF NEXT PAGE
T
GO TO 66001
PAGE 24
BOTTOM OF PRECEDING PAGE
C**** TYPE MESSAGE ON REAL TIME TYPEWRITER IF DESIRED
CALL
CALL HALT
ENCODE(130,66000,TYBUF) OUT(1),OUT(3),OUT(6),IT,OUT(4),OUT(5),OUT(2),(OUT(K),K=7,10)
IBY=0
CALL
CALL WRT218(TYBUF,IBY,IXI)
CALL
CALL READY
C7 PG. 23
66001
CONTINUE
GO TO 8195
C10 PG.25
C8 PG.12 .12
8190
CONTINUE
8191
CALL
CALL HALT
ENCODE(82, 5300, TYBUF) (OUT(K), K=2,10)
IBY = 0
CALL
CALL WRT218(TYBUF, IBY, IXI)
CALL
CALL READY


PAGE 25
C10 PG. 24
C**** MOVE MSG TOWARD FRONT OF FILE NCMSG(I)=NCMSG(I)-1
8195
NCMSGT=NCMSG(I)
F
IF(NCMSGT.LE.1)
T
GO TO 8205
DO 8200 J=2,NCMSGT
JJ=J-1
ICMSG1(I,JJ)=ICMSG1(I,J)
ICMSG3(I,JJ)=ICMSG3(I,J)
8200
ICMSG2(I,JJ)=ICMSG2(I,J)
8205
NCMSG(I)=NCMSGT-1
GO TO 8900
D20 PG.30
C13 PG. 11
8225
DAC(I)=NFMSG(I)*0.01
C**** MSG FILE EMPTY NCMSG(I)=0
F
IF(NCMSGT.EQ.0)
TOP OF NEXT PAGE
PAGE 26
T
GO TO 8400
C16 PG.28
BOTTOM OF PRECEDING PAGE
DAC(I) = 0.25
C**** IS IT TOO LATE TO TRANSMIT 1ST MSG
F
IF(IT.LE.ICMSG1(I,1))
T
GO TO 8325
C15 PG.27
IOUT=4
OUT(1)=I
OUT(2)=ICMSG1(I,1)
CALL
CALL UPKSPL(IONE,ISIX,ICMSG2(I,1),IPMSG2,TCMSG)
CALL
CALL UPKSPL(IONE,ITHREE,ICMSG3(I,1),IPMSG3,TCMSG2)
F
IF(.NOT.MSGTYP)
T
GO TO 66008
C14 PG.27
CALL
CALL HALT


PAGE
ENCODE(139,66007,TYBUF) OUT(1),OUT(3),IT,OUT(4),OUT(5),OUT(2),
TIME,(OUT(K),K=7,10)
IBY=0
CALL
CALL WRT218(TYBUF,IBY,IX(I)
CALL
CALL READY
C14 PG. 26
66008
CONTINUE
DAC(I) = -0.25
ICOMM(I)=1
IDENT=OUT(3)
ICLEAR = OUT(6)
C**** ISSUE CLEARANCE
GO TO 8139
A19 PG.11
C15 PG. 26
8325
CALL
CALL UPKSPL(ITWO,ITWO,ICMSG2(I,1),IPMSG2,TEMPS)
ICOMM(I)=TEMPS
GO TO 8135
A18 PG.11
C**** ARE THERE A/C THAT NEED TO BE BROUGHT INTO THE ACTIVE LIST
C**** COMPUTED GO TO BASED ON WHICH CONTROLLER
PAGE
C16 PG. 26
8400
GO TO (8425,8425,8500,8500,8900,8500,8900,8900,8900,8900) I
C17 PG.28
C17 PG.28
C18 PG.28
C18 PG.28
D20 PG.30
D20 PG.30
D20 PG.30
D20 PG.30
D20 PG.30
D20 PG.30
C**** ARE THERE A/C IN THE ARRIVAL QUEUE
C**** CONTROLLERS 1 AND 2 -- ARRIVAL CONTROLLERS
C17 PG. 28 ,28
8425
F
IF(IA(I).EQ.0)
T
GO TO 8900
D20 PG.30
C**** PLACE A/C AND CP1 IN CCHECK FOR CONFLICTC ARRAY --
ICA=ICA+1
ITEMP=IENROT(I,1)
IACTN1(ICA) = ITEMP
IACTN2(ICA)=1
CALL
CALL UPKSPL(IFIVE,IFIVE,ITRAF2(ITEMP),IPTRAF2,TEMP)
ITEMP=TEMP
IACTN3(ICA)=ISTARTA(ITEMP)
GO TO 8900
D20 PG.30
C**** ARE THERE A/C IN THE DEPARTURE QUEUE
C**** CONTROLLERS 3 AND 4 -- DEPARTURE CONTROLLERS
C18 PG. 28 ,28
8500
II=I-2



F
IF(ID(II).EQ.0)
T
GO TO 8900
D20 PG.30
F
IF(II.EQ.2.OR. ID(2).EQ.0)
T
GO TO 8550
INUM1=ITOQUE(1,1)
INUM2=ITOQUE(2,1)
CALL
CALL UPKSPL(ISIX,ISIX,ITRAF2(INUM1),IPTRAF2,TEMPS)
CALL
CALL UPKSPL(ISIX,ISIX,ITRAF2(INUM2),IPTRAF2,TEMPS2)
F
IF(TEMPS.GT.TE MPS2)
T
GO TO 8900
D20 PG.30
PAGE 30
C**** PLACE A/C AND CP1 IN CCHECK FOR CONFLICT ARRAY C --
8550
ICA=ICA+1
ITEMP=ITOQUE(II,1)
IACTN1(ICA)=ITEMP
IACTN2(ICA)=2
CALL
CALL UPKSPL(IFIVE,IFIVE,ITRAF2(ITEMP),IPTRAF2,TEMP)
ITEMP=TEMP
IACTN3(ICA)=ISTARTA(ITEMP)
PG. 11 ,25 ,28 ,28 ,28 ,28
8900
CONTINUE
C**** RETURN TO BOTTOM OF OPERATE
ICNGO = 1
55
CONTINUE
PG. 7
90050
RETURN
END

```
SUBROUTINE CNTRLAC

LOGICAL LDISI. LDISO

COMMON / REALTIM/ ADC(32).DAC(64).LDISI(108).LDISO(196).
NOPER.NHOLD.NRESET.NTERM.NPRINT.NREAD

COMMON/TABLES/
    INTEG(80).       TABLE(075).      LOGIC(20)

LOGICAL LOGIC

*****
* * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN
*****

COMMON/MAJOR1/
     ACLS(50).        ALTOPT(3.7.2).   ALTSKD(9).
     ARRMNA.          ARRMXA.          BOUND.
     CGLIDE.          CILS.            CRA20.
     CRA30.           CRB20.           CRB30.
     DGRTD.           DTR.             EHOLDO,
     ENROTA(2).       FTNM.            G.
     GLIDE(2).        HOLDSEP.         IARRN.
     IARRS.           IART(11).        IBF(110).
     ICHKPT(40).      ICNTRA1(85).     ICNTRA2(85).
     IDELT.           IDEPN.           IDEPS.
     IDROUTE(10).     IHPATHE.         IHPATHS.
     INAV(10.5).      IOPTN(55).       IPAPATH(4.5).
     IPASKED(4.7).    IPCHKPT(4.6).    IPCNTR1(4.6).
     IPCNTR2(4.2).    IPDRT(4.5).      IPERFR(3.7).
     IPERFS(4.8).     IPERFT(3).       IPETA1(4.4).
     IPETA2(4.6).     IPFTR(3).        IPMSG2(4.6).
     IPMSG3(4.3).     IPNPATH(4.7).    IPNSITE(4.4).
     IPOPL(2).        IPOPPA.          IPOPR(2).
     IPOPTN(4.9).     IPSKED(4.8).     IPTAS(4.5)

COMMON/MAJOR2/
     IPTRAF1(4.4).    IPTRAF2(4.9).    ISDST(17).
     ISKDPU.          ISKED(85).       ISTARTA(21).
     ITANG(2.2).      ITRAF1(102).     ITRAF2(102).
     MAXK.            MF.
     NALTSKD.         NCP.             NDEPR.
     NNAV(5).         NNPATH(50).      NOUT(20).
     NSPDW.           OUT(26).         PI.
     PPMX(3).         PPMY(3).         RCP(2).
     RTD.             RWYX(4.2).       RWYY(4.2).
     RWYZ(2).         SHOLDD.          SPOSTD(9.2).
     SQX(6).          SQY(6).          TRIX(5).
     TRIY(5).         XA(2).           XC(2).
     XDILS(2).        XDIST(2).        XFINAL(2).
     XRPOS(2).        YA(2).           YC(2).
     YRPOS(2).        RAD(3).          SIG(3).          IARG
   . IETCOOR(36).     IPETC(4.2).      IACFPER(3)

*****

COMMON/TOES/ IONE.ITWO.ITHREE.IFOUR.IFIVE.ISIX.ISEVEN.IEIGHT.ININE
.ITEN

*****
* * * REAL-TIME MAIN COMMON IN OVERLAY(4.0) AND ALL SECONDARY OVERLAYS A
* * *      PRIMARY SUBROUTINES OF (4.0)
*****

COMMON/RTMAIN1/
   ASKSPD(35.11).ASPACE1(4.9).ASPACE2(4.9).ASPACE3(4.9).
     CETA.            CONFLCT.         DELH(35).
     HDBEG(35).       HDOWN.           IA(2).
     IAC.             IACTN1(35).      IACTN2(35).
  IACTN3(35). IASKED(35.11). IATAS(35.2).ICNGO.
     IATRAF(35.26).   ICA.             ICACT.
     ICLASS.          ICMSG1(8.5).     ICMSG2(8.5).
     ICMSG3(8.5).     ICOMM(10).       ID(2).
     IDELAY.          IENROT(2.20).    IETAS(35.2).
     IETA1(36.10).    IETA2(36.10).    IE360(60).
     IFCMSG1(8.25).   IFCMSG2(8.25).   IFCMSG3(8.25).
     IHOWN.           IOBETA.          IOEETA.
     IOPATH.          IPASSAC.         ISEQ.
     ISEQUE(3.16).    ISW4.            ITOQUE(2.20).
     ITRY.            IWHICH.          KNTP.
     L.               LEMG(5).         NASPACE(4).
     NCMSG(10).       NEMG.            NETA(36).
     NFMSG(10).       NHOLDP(35).      NNA(35).
     NSEQ.            OLDR(35)

COMMON/RTMAIN2/
     OTHAA.           OTHETA.          OTHOA.
     OTHSETA.         RESTOR(35).      ROLL.
     T(35).           TETAE.           TETAL.
     TETAO.           TIME.            V1.
     V2.              XOWN.            XPL(35).
     YOWN.            YPL(35).         ZOWN
   . R9LX(2).         R9LY(2).         R9RX(2).         R9RY(2).
     R15X(2).         R15Y(2).         R21X(2).         R21Y(2).
     XNREF(4).        YNREF(4)

*****
*     SPECIAL COMMON FOR CNTRLAC
*****

COMMON/RTMAIN3/
     ALTLIM.          DALT.            DELAYT.
     HOLDTIM.         HT1.             HT2.
     IACTPT.          ICNT.            ICONTRL.
     IDENT.           IDVERG.          IETT.            IDEST.
     IHEM.            ILEG.            INEXT.
     IRCLR.           IS.              ITEMP.
     ITEMPA(10).      ITEMPS.          ITRYDVG.
     ITRYEH.          ITRYSH.          ITRYSPD.
     ITRYVEC.         ITYPCL.          ITYPE.
     IVARSPD.         IVECT2.          IWCHAD.
     IWHCH.           IWHCNTR.         IWHSPD.
     J.               JGO.             JJ.
     JJJ.             JL.              JTEMPA(10).
     K.               KGO.             KALTOP.
     KK.              K3.              LIND.
     NCLRD.           NOM.             SAVE(35).
     SAVE2(35).       SPDSAV.          TETAOM.
     TMSGA(6).        TMSGA2(3).       RCFNM(2)
```



```
COMMON/RTMAIN4/
 A. ANGLE.     ANG2.      B.         I.         IAALT.     IANG.
    IAT.       ICLP1.     ICONTR.    ICP.       IDT.       IETT2.
    IGO.       IH.        ILEGM1.    IND.       INDR.      IPA.
    IPD.       IRSEP.     IRT.       ISA.       ISD.       ISEQM1.
    ISUBSPD.   ITEMP2.    IWAC.      IWETA.     IWHP1.     IWID.
    JJM1.      JJP1.      JOE.       JP1.       JS.        JTEMP.
    LL.        LLL.       LLLP3.     M.         MM.        MMM.
    MTEMP.     NCMSGT.    NFMSGT.    NTEMP.     N360.      RATE.
    RC.        RDESC.     R1.        SEPCOR.    SPEED.     TAFIX.
    TEMETA.    TEMGAT.    TEMOFF.    TEMONN.    TEMP.      TEMPA3(10).
    TEMSEP.    TTH.       TXA.       TXC.       TXI.       TYA.
    TYC.       TYI.       XB.        XCLR2.     XI.        XLID.
    XLTT2.     XMSGL2.    YB.        YI

*****
* * * DIM. + TYPE  FOR  CNTRLAC

DIMENSION
    MSG(20)  . ALTA(2.2) . SEPP(2)

* * * TABLE. INTEG. AND LOGIC  EQIV. FOR  CNTRLAC

EQUIVALENCE (INTEG(06). NFMSGD). (INTEG(07). KRET).
     (INTEG(18). ICNTF)

EQUIVALENCE
    (INTEG(08).        INC)      .(INTEG(11).          IO)
   .(INTEG(12).       IOUT)      .(INTEG(30).       MAXAC)
   .(INTEG(33).       NCOM)      .(INTEG(40).           IT)
   .(INTEG(41).      NSCAN)      .(INTEG(46).       MSG(1))
   .(TABLE(03).ALTA(1.1))        .(TABLE(11).        TEMPS)
   .(TABLE(16).       DELT)      .(TABLE(22).        HOLDT)
   .(TABLE(23).      PASEP)      .(TABLE(24).        POSEP)
   .(TABLE(38).      SASEP)      .(TABLE(40).         SCLA)
   .(TABLE(41).       SCLE)      .(TABLE(42).         SCLN)
   .(TABLE(43). SEPP(1))         .(TABLE(12).VECERR)

*****
* * * DIMENSIONS AND EQUIVALENCES FOR CNTRLAC
*****

DIMENSION DL(3)

DIMENSION
     ATRAF(35.26).   IATRAF1(35).    ATRAF2(35).     ATRAF3(35).
     ATRAF4(35).     ATRAF5(35).     ATRAF6(35).     IATRAF7(35).
     IATRAF8(35).    IATRAF9(35).    IATRF11(35).    IATRF12(35).
     IATRF16(35).    IATRF17(35).     ATRF18(35).    ATRF19(35).
     ATRF20(35).                     IPERFR6(3).     IPERFR7(3).
     IPERFS1(4).     IPERFS2(4).     IPERFS5(4).     IPERFS6(4).
     IPERFS7(4).     IPERFS8(4).     TEMPA(10).
     SEQUE(3.16).    ISAC(3).        ISIDE(3).
     ISCLAS(3).      TARR(3).        SVEL(3).        RA20(3).
     RA30(3).        RB20(3).        RB30(3).        TCP(3).
     TOFF(3).        TONN(3).        TGATE(3).       ANG(3)

EQUIVALENCE  (DL(1).ISEQUE(1.15))

EQUIVALENCE
     (IATRAF(1.1).   ATRAF(1.1)).    (IATRAF(1.1).   IATRAF1(1)).
     (ATRAF(1.2).    ATRAF2(1)).     (ATRAF(1.3).    ATRAF3(1)).
     (ATRAF(1.4).    ATRAF4(1)).     (ATRAF(1.5).    ATRAF5(1)).
     (ATRAF(1.6).    ATRAF6(1)).     (IATRAF(1.7).   IATRAF7(1)).
     (IATRAF(1.8).   IATRAF8(1)).    (IATRAF(1.9).   IATRAF9(1)).
     (IATRAF(1.11).  IATRF11(1)).    (IATRAF(1.12).  IATRF12(1)).
     (IATRAF(1.16).  IATRF16(1)).    (IATRAF(1.17).  IATRF17(1)).
     (IATRAF(1.18).   ATRF18(1)).    (ATRAF(1.19).   ATRF19(1)).
     (ATRAF(1.20).   ATRF20(1)).
     (IPERFR(1.6).   IPERFR6(1)).    (IPERFR(1.7).   IPERFR7(1)).
     (IPERFS(1.1).   IPERFS1(1)).    (IPERFS(1.2).   IPERFS2(1)).
     (IPERFS(1.5).   IPERFS5(1)).    (IPERFS(1.6).   IPERFS6(1)).
     (IPERFS(1.7).   IPERFS7(1)).    (IPERFS(1.8).   IPERFS8(1)).
     (TMSGA(1).      XDENT).         (TMSGA(2).      XMSGL).
     (TMSGA(3).      XLTT).          (TMSGA(4).      XCLEAR).
     (TMSGA(5).      XITYPE).        (TEMPA(1).      ITEMPA(1)).
     (ISEQUE(1.1).   SEQUE(1.1)).    (ISEQUE(1.1).   ISAC(1)).
     (ISEQUE(1.2).   ISIDE(1)).      (ISEQUE(1.3).   ISCLAS(1)).
     (SEQUE(1.4).    TARR(1)).       (SEQUE(1.5).    SVEL(1))

* * * CONTINUATION OF CNTRLAC EQUIVALENCES

EQUIVALENCE
     (SEQUE(1.6).    RA20(1)).       (SEQUE(1.7).    RA30(1)).
     (SEQUE(1.8).    RB20(1)).       (SEQUE(1.9).    RB30(1)).
     (SEQUE(1.10).   TCP(1)).        (SEQUE(1.11).   TOFF(1)).
     (SEQUE(1.12).   TONN(1)).       (SEQUE(1.13).   TGATE(1)).
     (SEQUE(1.14).   ANG(1))

DIMENSION
     IATRF21(35).    IATRF22(35).    IATRF23(35).    IATRF24(35).
     IPERFR2(3).     IPERFR3(3).     IPERFR4(3).     IPERFR5(3).
     IOPTION(9).     OPTION(9).
     FETA(3).        IPERFS3(4).     ATRF26(35)

DIMENSION   IPERFR1(3)
EQUIVALENCE   (IPERFR(1.1). IPERFR1(1))

EQUIVALENCE
     (IATRAF(1.21). IATRF21(1)).     (IATRAF(1.22). IATRF22(1)).
     (IATRAF(1.23). IATRF23(1)).     (IATRAF(1.24). IATRF24(1)).
     (TMSGA(6). XETA).               (TMSGA2(1). XMINFO1).
     (TMSGA2(2). XMINFO2).           (TMSGA2(3). XMINFO3).
     (ISEQUE(1.16). FETA(1)).        (IPERFS(1.3). IPERFS3(1)).
     (IPERFR(1.2). IPERFR2(1)).      (IPERFR(1.3). IPERFR3(1)).
     (IPERFR(1.4). IPERFR4(1)).      (IPERFR(1.5). IPERFR5(1)).
     (IATRAF(1.26). ATRF26(1))
```

```
EQUIVALENCE
   (RESTOR(01). TETA).            (RESTOR(02). NUMPATH).
   (RESTOR(03). OALT).            (RESTOR(04). AALT).
   (RESTOR(05). ETASEP).          (RESTOR(06). FVEL).
   (RESTOR(07). OSPD).            (RESTOR(08). TX).
   (RESTOR(09). TOTD).            (RESTOR(10). NSEGM).
   (RESTOR(11). OST1).            (RESTOR(12). OST2).
   (RESTOR(13). OST3).            (RESTOR(14). ISUBCK).
   (RESTOR(15). ISUBOP).          (RESTOR(16). ICACT2).
   (RESTOR(17). STEMP).           (RESTOR(18). ISUBN).
   (RESTOR(19). ISUBO).           (RESTOR(20). SEP).
   (RESTOR(21). XLA).             (RESTOR(22). XLB).
   (RESTOR(23). IRSPCON).
   (RESTOR(24). IOALT. IOPTION(1). OPTION(1)).
   (RESTOR(25). IOSPD).           (RESTOR(26). IVECT).
   (RESTOR(27). IVECOP).          (RESTOR(28). IOVERG2).
   (RESTOR(29). IALTOP).          (RESTOR(30). ISPDOP).
   (RESTOR(31). ISHOP).           (RESTOR(32). IEHOP)
 . (RESTOR(33). OSPD)
 . (RESTOR(34). TX3).             (RESTOR(35). INDSPD)

*****
* * * DIMENSIONS FOR CNTRLAC
*****

DIMENSION
     TYBUF(16)

DIMENSION FSS(16)
LOGICAL FSS
EQUIVALENCE  (LDISI(33). FSS(1))

GO TO (8000. 21422. 21710. 24505. 25850. 25950. 26100
 .      12800
 .   ) ICNGO
```

A1	PG.6
J18	PG.77
K6	PG.80
M10	PG.98
P7	PG.110
P10	PG.112
P17	PG.115
C7	PG.30

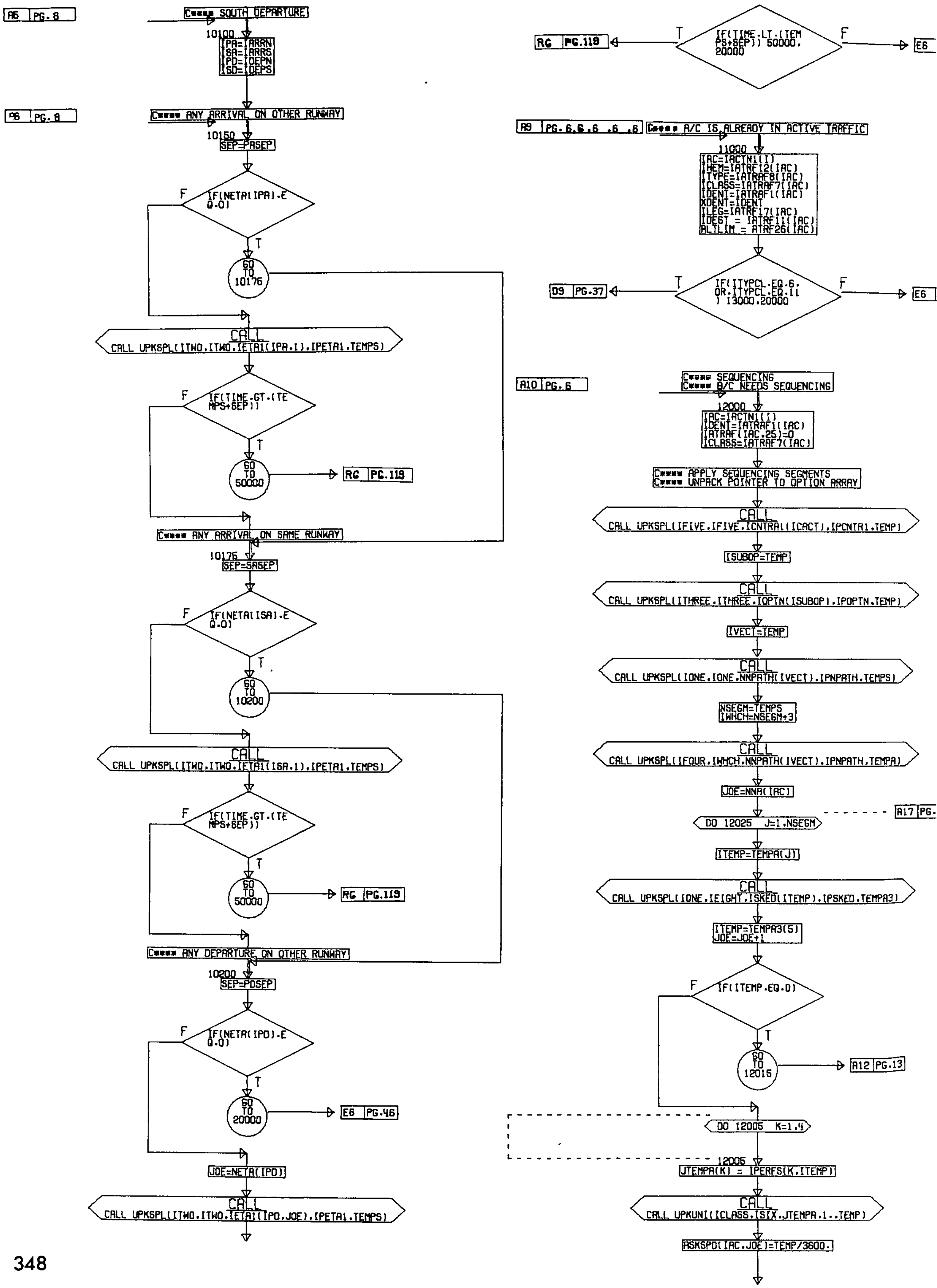
```
1  PG.5
8000
   CONTINUE
NFMSGD = 0
C**** CHECK TO SEE IF CONTROLLER ACTION IS NEEDED
ITEMP = INC-NCOM-ICNTF
IF(ITEMP.GT.ICA)   F / T
   T: GO TO 90050   -> A7  PG.119
C**** I = FRAME COUNTER - TRACKING FRAMES -COMMUNICATION FRAMES
C**** I IS AN INDEX FOR THE ARRAY OF A/C THAT NEED CONTROLLER ACTION
I=ITEMP
ICACT=IACTN3(I)
ITYPCL=IACTN2(I)
*****
***** UNPACK TOTAL DISTANCE BEING CHECKED AND CONTROLLER RESPONSIBLE
*****    FOR ACTION
*****
CALL
CALL UPKSPL(IONE.ITWO.ICNTRA2 (ICACT).IPCNTR2.TEMPA)
ICONTRL=TEMPA(1)
TOTD=TEMPA(2)
C**** COMPUTED GO TO BASED ON TYPE CHECK NEEDED
GO TO (10000.10000.11000.12000.11000.11000.11000.50000.15000.
15000.11000).ITYPCL
```

A2	PG.6
A2	PG.6
A9	PG.11
A10	PG.11
A9	PG.11
A9	PG.11
A9	PG.11
A6	PG.119
D16	PG.42
D19	PG.44
A9	PG.11

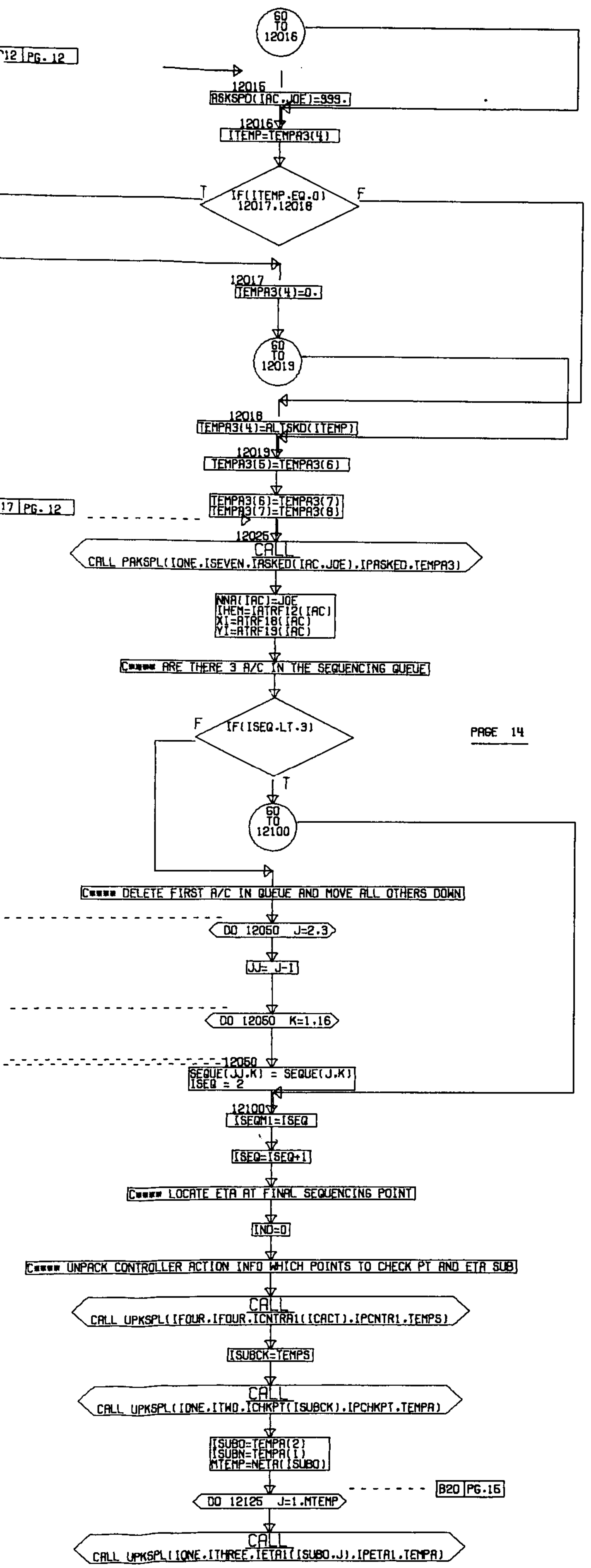
```
A2  PG.6  .6   C**** A/C IS WAITING IN QUEUE FOR CONTROLLER ACTION
10000
   INEXT=IACTN1(I)
```

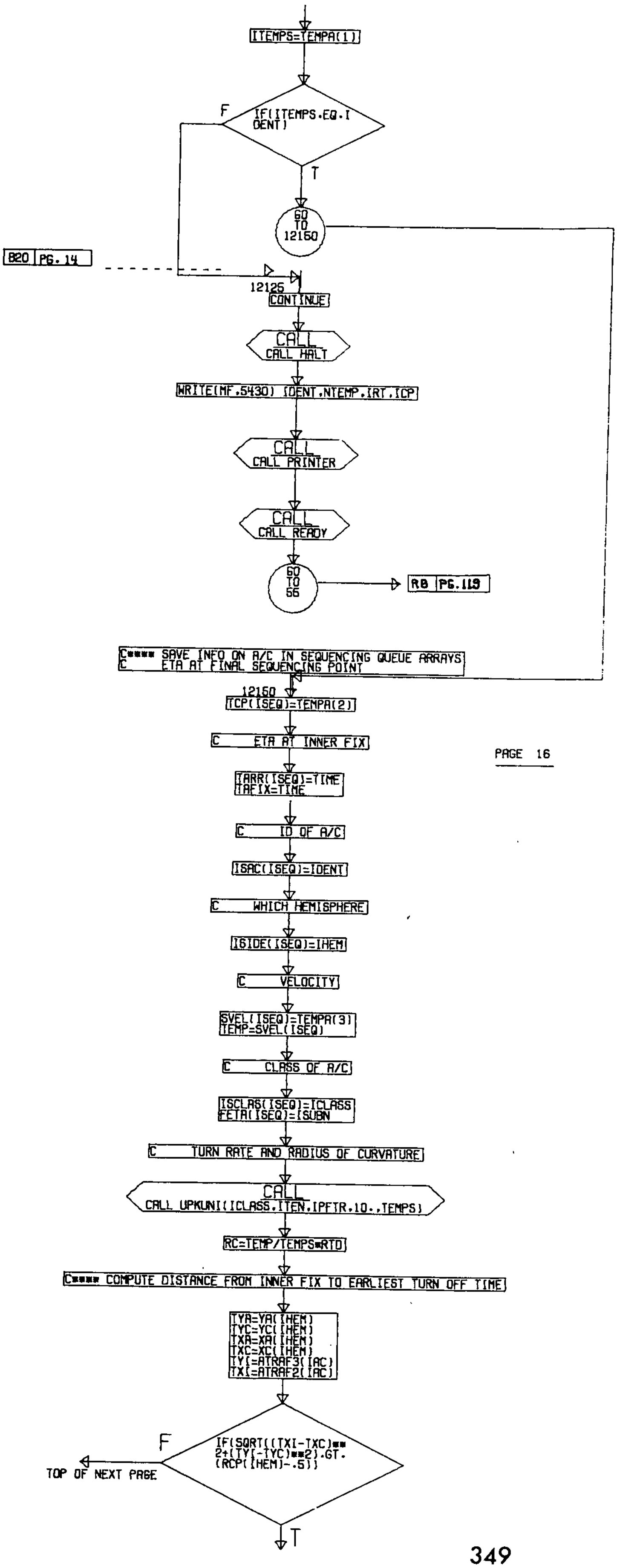


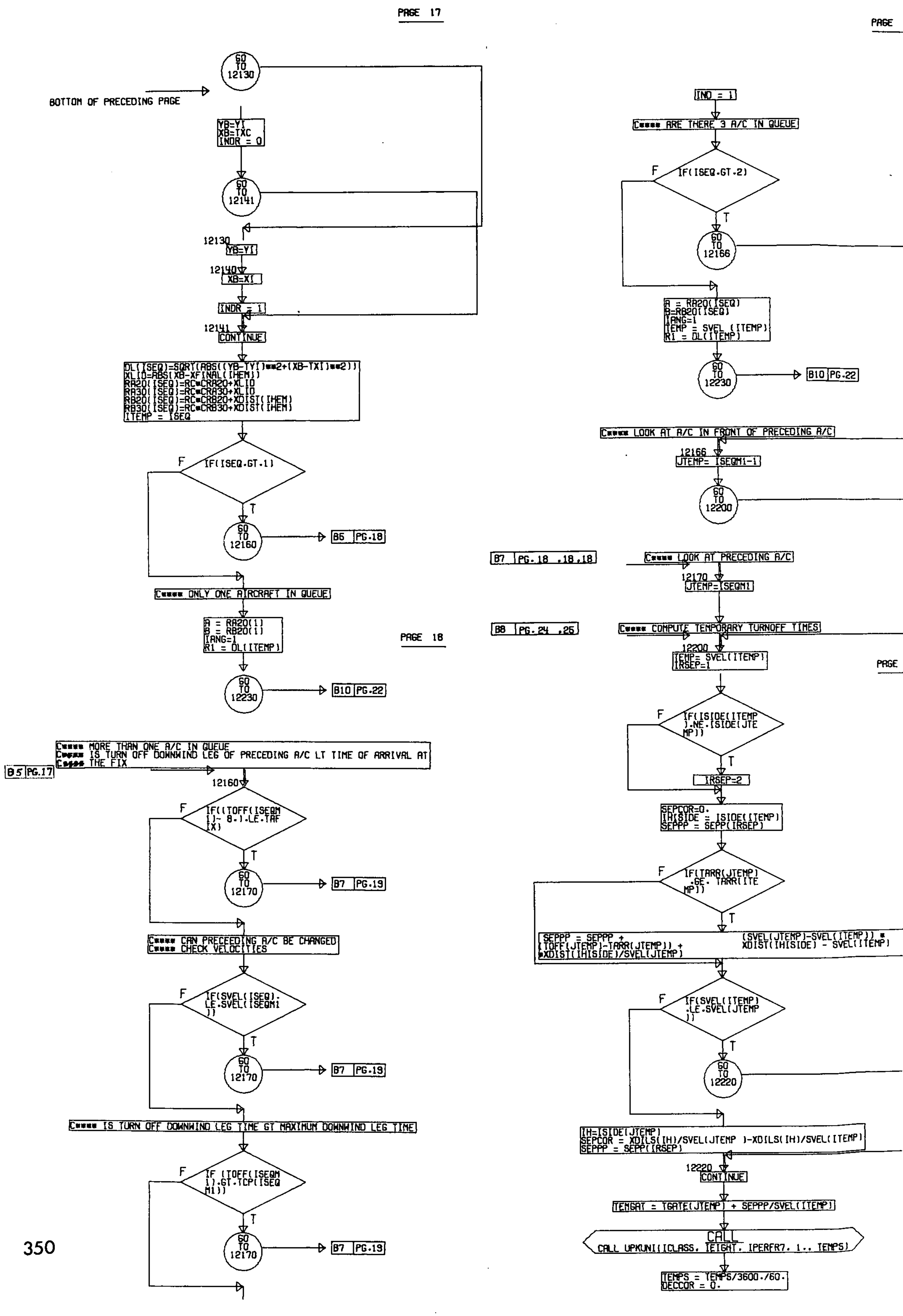
```
CALL
CALL UPKSPL(IONE.IONE.ITRAF1(INEXT).IPTRAF1.XDENT)
IDENT=XDENT
CALL
CALL UPKSPL(IONE.IONE.ITRAF2(INEXT).IPTRAF2.OSPD)
OSPD=OSPD/3600.
CALL
CALL UPKSPL(ITHREE.IFOUR.ITRAF2(INEXT).IPTRAF2.TEMPA)
ICLASS=TEMPA(1)
ITYPE=TEMPA(2)
CETA=0.
XMSGL=MSG(1)
TETAO=TIME     +XMSGL+1.
ILEG=1
CALL
CALL UPKSPL(ISEVEN.ININE.ITRAF2(INEXT).IPTRAF2.TEMPA)
IDEST=TEMPA(1)
IHEM=TEMPA(2)
ALTLIM= TEMPA(3)
CALL
CALL UPKSPL(IFOUR.IFOUR.ITRAF1(INEXT).IPTRAF1.OALT)
IF(IDEST.EQ.2)   F / T
   T: GO TO 10050   -> A3  PG.8
CALL
CALL UPKUNI(ICLASS.ISIX.IPERFS1.1..OSPD)
OSPD=OSPD/3600.
A3  PG.7
10050
   CONTINUE
10075
   CONTINUE
C**** IF THIS A/C IS A DEPARTURE. A RUNWAY CHECK MUST BE DONE BEFORE
C****    THE GENERAL CONFLICT CHECK
IF(IDEST.NE.1)   F / T
   T: GO TO 20000   -> E6  PG.46
C**** RUNWAY CHECK
C**** IS THE DEPARTURE FROM THE NORTH OR SOUTH
IF(IHEM.EQ.2)   F / T
   T: GO TO 10100   -> A6  PG.9
C**** NORTH DEPARTURE
IPA=IARRS
ISA=IARRN
IPD=IDEPS
ISD=IDEPN
GO TO 10150   -> A6  PG.9
```



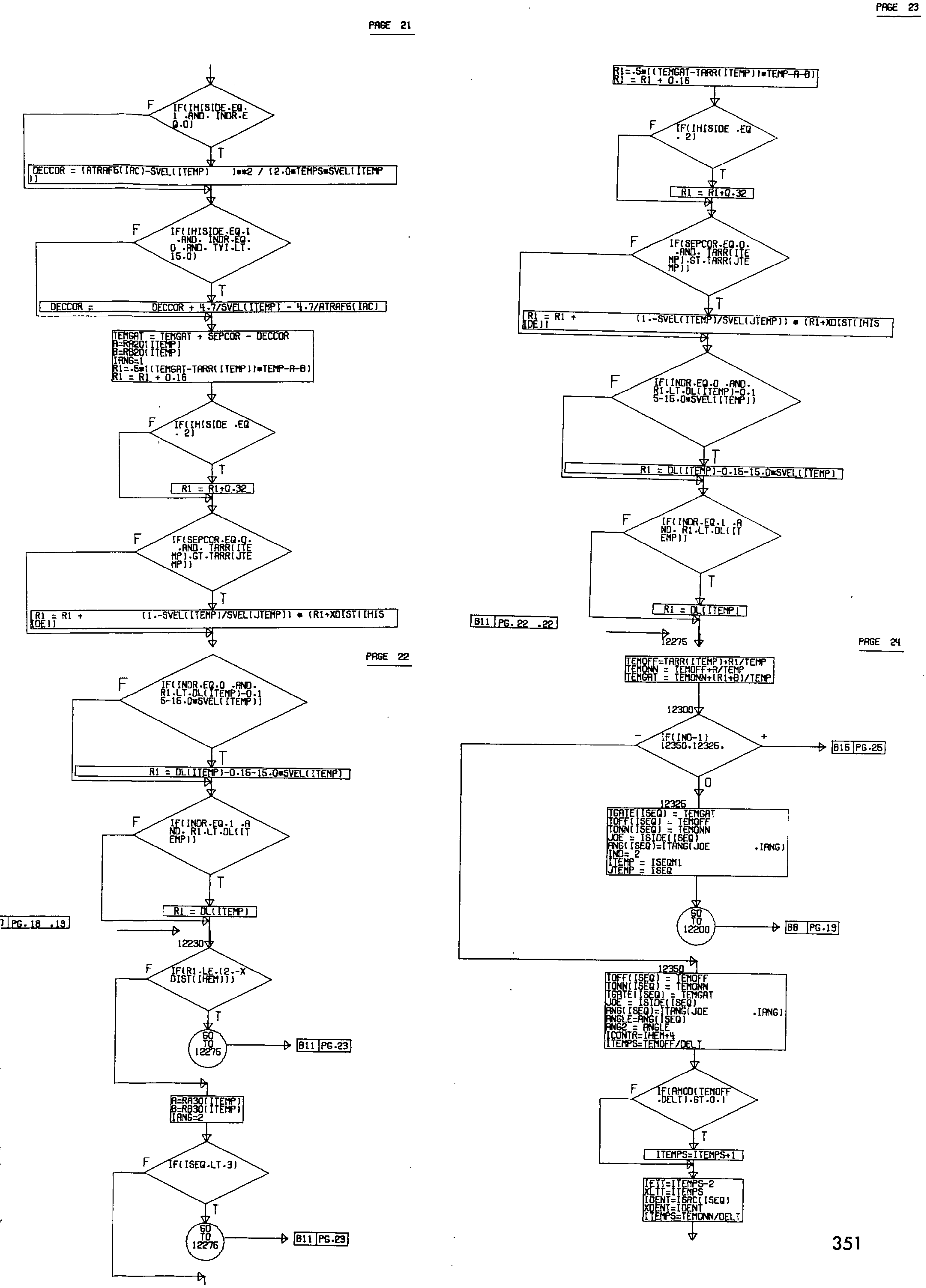
A6 PG.8
C**** SOUTH DEPARTURE
10100
IPA=IARRN
ISA=IARRS
IPD=IDEPN
ISD=IDEPS
A6 PG.8
C**** ANY ARRIVAL ON OTHER RUNWAY
10150
SEP=PASEP
IF(NETA(IPA).EQ.0)
F
T
GO TO 10175
CALL
CALL UPKSPL(ITWO,ITWO,IETA1(IPA,1),IPETA1,TEMPS)
IF(TIME.GT.(TEMPS+SEP))
F
T
GO TO 50000
RG PG.119
C**** ANY ARRIVAL ON SAME RUNWAY
10175
SEP=SASEP
IF(NETA(ISA).EQ.0)
F
T
GO TO 10200
CALL
CALL UPKSPL(ITWO,ITWO,IETA1(ISA,1),IPETA1,TEMPS)
IF(TIME.GT.(TEMPS+SEP))
F
T
GO TO 50000
RG PG.119
C**** ANY DEPARTURE ON OTHER RUNWAY
10200
SEP=PDSEP
IF(NETA(IPD).EQ.0)
F
T
GO TO 20000
E6 PG.46
JOE=NETA(IPD)
CALL
CALL UPKSPL(ITWO,ITWO,IETA1(IPD,JOE),IPETA1,TEMPS)
IF(TIME.LT.(TEMPS+SEP)) 50000,20000
T
F
RG PG.119
E6
A9 PG.6,6,6,6,6
C**** A/C IS ALREADY IN ACTIVE TRAFFIC
11000
IAC=IACTN1(I)
IHEM=IATRF12(IAC)
ITYPE=IATRAF8(IAC)
ICLASS=IATRAF7(IAC)
IDENT=IATRAF1(IAC)
XDENT=IDENT
ILEG=IATRF17(IAC)
IDEST = IATRF11(IAC)
ALTLIM = ATRF26(IAC)
IF(ITYPCL.EQ.6.OR.ITYPCL.EQ.11) 13000,20000
T
F
D9 PG.37
E6
A10 PG.6
C**** SEQUENCING
C**** A/C NEEDS SEQUENCING
12000
IAC=IACTN1(I)
IDENT=IATRAF1(IAC)
IATRAF(IAC,25)=0
ICLASS=IATRAF7(IAC)
C**** APPLY SEQUENCING SEGMENTS
C**** UNPACK POINTER TO OPTION ARRAY
CALL
CALL UPKSPL(IFIVE,IFIVE,ICNTRA(ICACT),IPCNTR1,TEMP)
ISUBOP=TEMP
CALL
CALL UPKSPL(ITHREE,ITHREE,IOPTN(ISUBOP),IPOPTN,TEMP)
IVECT=TEMP
CALL
CALL UPKSPL(IONE,IONE,NNPATH(IVECT),IPNPATH,TEMPS)
NSEGM=TEMPS
IWHCH=NSEGM+3
CALL
CALL UPKSPL(IFOUR,IWHCH,NNPATH(IVECT),IPNPATH,TEMPA)
JOE=NNA(IAC)
DO 12025 J=1,NSEGM
A17 PG.
ITEMP=TEMPA(J)
CALL
CALL UPKSPL(IONE,IEIGHT,ISKED(ITEMP),IPSKED,TEMPA3)
ITEMP=TEMPA3(5)
JOE=JOE+1
IF(ITEMP.EQ.0)
F
T
GO TO 12015
A12 PG.13
DO 12005 K=1,4
12005
JTEMPA(K) = IPERFS(K,ITEMP)
CALL
CALL UPKUNI(ICLASS,ISIX,JTEMPA,1,,TEMP)
ASKSPD(IAC,JOE)=TEMP/3600.


GO TO 12016
12 PG. 12
12015
RASKSPD(IAC.JOE)=999.
12016
ITEMP=TEMPA3(4)
T
IF(ITEMP.EQ.0) 12017.12018
F
12017
TEMPA3(4)=0.
GO TO 12019
12018
TEMPA3(4)=ALTSKD(ITEMP)
12019
TEMPA3(5)=TEMPA3(6)
TEMPA3(6)=TEMPA3(7)
TEMPA3(7)=TEMPA3(8)
17 PG. 12
12025
CALL
CALL PAKSPL(IONE.ISEVEN.IASKED(IAC.JOE).IPASKED.TEMPA3)
NNA(IAC)=JOE
IHEM=IATRF12(IAC)
XI=ATRF18(IAC)
YI=ATRF19(IAC)
C**** ARE THERE 3 A/C IN THE SEQUENCING QUEUE
F
IF(ISEQ.LT.3)
PAGE 14
T
GO TO 12100
C**** DELETE FIRST A/C IN QUEUE AND MOVE ALL OTHERS DOWN
DO 12050 J=2.3
JJ= J-1
DO 12050 K=1.16
12050
SEQUE(JJ.K) = SEQUE(J.K)
ISEQ = 2
12100
ISEQM1=ISEQ
ISEQ=ISEQ+1
C**** LOCATE ETA AT FINAL SEQUENCING POINT
IND=0
C**** UNPACK CONTROLLER ACTION INFO WHICH POINTS TO CHECK PT AND ETA SUB
CALL
CALL UPKSPL(IFOUR.IFOUR.ICNTRA1(ICACT).IPCNTR1.TEMPS)
ISUBCK=TEMPS
CALL
CALL UPKSPL(IONE.ITWO.ICHKPT(ISUBCK).IPCHKPT.TEMPA)
ISUBO=TEMPA(2)
ISUBN=TEMPA(1)
MTEMP=NETA(ISUBO)
DO 12125 J=1.MTEMP
B20 PG.15
CALL
CALL UPKSPL(IONE.ITHREE.IETA1(ISUBO.J).IPETA1.TEMPA)

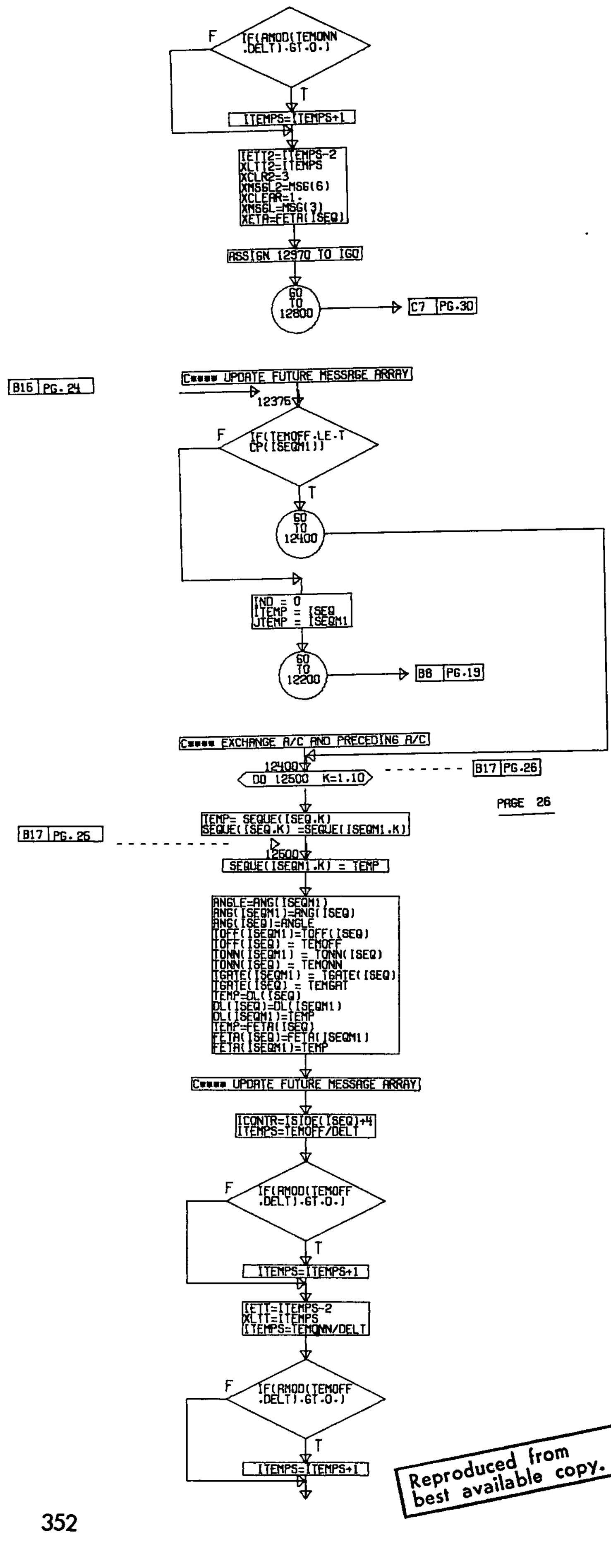

ITEMPS=TEMPA(1)
F
IF(ITEMPS.EQ.I DENT)
T
GO TO 12150
B20 PG. 14
12125
CONTINUE
CALL
CALL HALT
WRITE(MF.5430) IDENT.NTEMP.IRT.ICP
CALL
CALL PRINTER
CALL
CALL READY
GO TO 55
R8 PG.119
C**** SAVE INFO ON A/C IN SEQUENCING QUEUE ARRAYS
C ETA AT FINAL SEQUENCING POINT
12150
ICP(ISEQ)=TEMPA(2)
C ETA AT INNER FIX
IARR(ISEQ)=TIME
IAFIX=TIME
PAGE 16
C ID OF A/C
ISAC(ISEQ)=IDENT
C WHICH HEMISPHERE
ISIDE(ISEQ)=IHEM
C VELOCITY
SVEL(ISEQ)=TEMPA(3)
TEMP=SVEL(ISEQ)
C CLASS OF A/C
ISCLAS(ISEQ)=ICLASS
FETA(ISEQ)=ISUBN
C TURN RATE AND RADIUS OF CURVATURE
CALL
CALL UPKUNI(ICLASS.ITEN.IPFTR.10..TEMPS)
RC=TEMP/TEMPS*RTD
C**** COMPUTE DISTANCE FROM INNER FIX TO EARLIEST TURN OFF TIME
TYA=YA(IHEM)
TYC=YC(IHEM)
TXA=XA(IHEM)
TXC=XC(IHEM)
TYI=ATRAF3(IAC)
TXI=ATRAF2(IAC)
F
IF(SQRT((TXI-TXC)**2+(TYI-TYC)**2).GT.(RCP(IHEM)-.5))
TOP OF NEXT PAGE
T

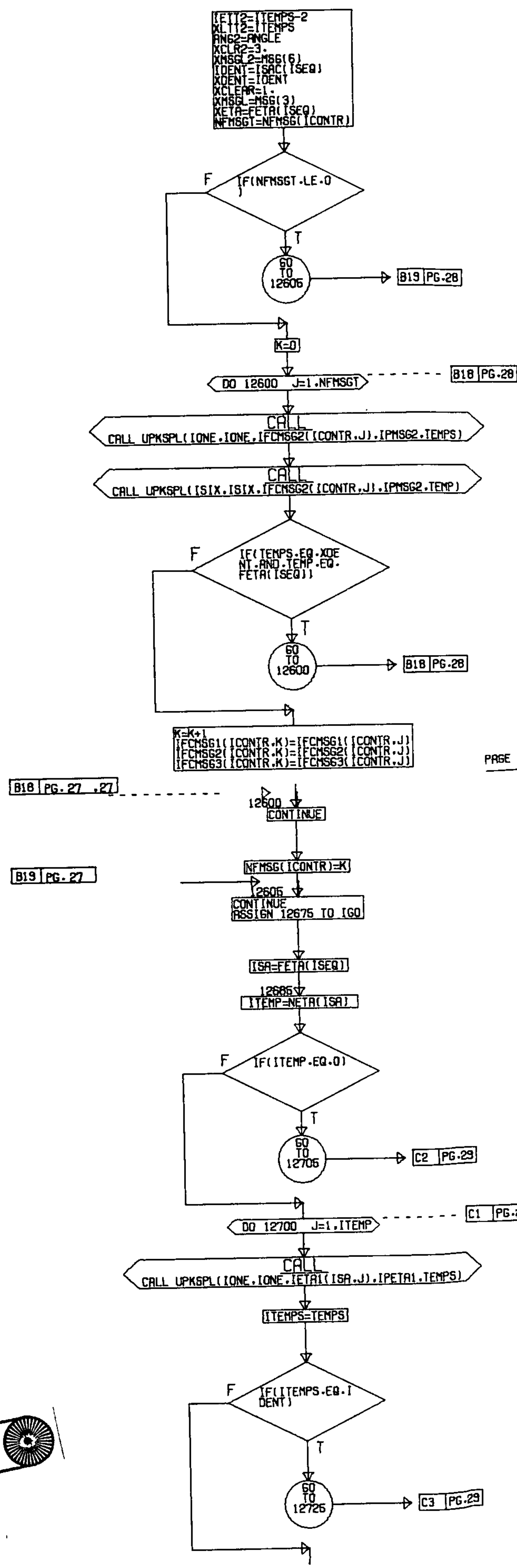


PAGE 17
GO TO 12130
BOTTOM OF PRECEDING PAGE
YB=YI
XB=TXC
INDR = 0
GO TO 12141
12130 YB=YI
12140 XB=XI
INDR = 1
12141 CONTINUE
DL(ISEQ)=SQRT(ABS((YB-TYI)**2+(XB-TXI)**2))
XLID=ABS(XB-XFINAL(IHEM))
RA20(ISEQ)=RC*CRA20+XLID
RA30(ISEQ)=RC*CRA30+XLID
RB20(ISEQ)=RC*CRB20+XDIST(IHEM)
RB30(ISEQ)=RC*CRB30+XDIST(IHEM)
ITEMP = ISEQ
IF(ISEQ.GT.1)
F
T
GO TO 12160
B6 PG.18
C**** ONLY ONE AIRCRAFT IN QUEUE
A = RA20(1)
B = RB20(1)
IANG=1
R1 = DL(ITEMP)
GO TO 12230
B10 PG.22
PAGE 18
C**** MORE THAN ONE A/C IN QUEUE
C**** IS TURN OFF DOWNWIND LEG OF PRECEDING A/C LT TIME OF ARRIVAL AT
C**** THE FIX
B5 PG.17
12160
IF((TOFF(ISEQM1)- 8.).LE.TAFIX)
GO TO 12170
B7 PG.19
C**** CAN PRECEEDING A/C BE CHANGED
C**** CHECK VELOCITIES
IF(SVEL(ISEQ).LE.SVEL(ISEQM1))
GO TO 12170
B7 PG.19
C**** IS TURN OFF DOWNWIND LEG TIME GT MAXIMUM DOWNWIND LEG TIME
IF (TOFF(ISEQM1).GT.TCP(ISEQM1))
GO TO 12170
B7 PG.19
PAGE
IND = 1
C**** ARE THERE 3 A/C IN QUEUE
IF(ISEQ.GT.2)
GO TO 12166
A = RA20(ISEQ)
B=RB20(ISEQ)
IANG=1
TEMP = SVEL (ITEMP)
R1 = DL(ITEMP)
GO TO 12230
B10 PG.22
C**** LOOK AT A/C IN FRONT OF PRECEDING A/C
12166
JTEMP= ISEQM1-1
GO TO 12200
B7 PG.18 ,18,18
C**** LOOK AT PRECEDING A/C
12170
JTEMP=ISEQM1
B8 PG.24 ,25
C**** COMPUTE TEMPORARY TURNOFF TIMES
12200
TEMP= SVEL(ITEMP)
IRSEP=1
PAGE
IF(ISIDE(ITEMP).NE.ISIDE(JTEMP))
IRSEP=2
SEPCOR=0.
IHISIDE = ISIDE(ITEMP)
SEPPP = SEPP(IRSEP)
IF(TARR(JTEMP).GE. TARR(ITEMP))
SEPPP = SEPPP + (SVEL(JTEMP)-SVEL(ITEMP)) *
(TOFF(JTEMP)-TARR(JTEMP)) + XDIST(IHISIDE) - SVEL(ITEMP)
*XDIST(IHISIDE)/SVEL(JTEMP)
IF(SVEL(ITEMP).LE.SVEL(JTEMP))
GO TO 12220
IH=ISIDE(JTEMP)
SEPCOR = XDILS(IH)/SVEL(JTEMP)-XDILS(IH)/SVEL(ITEMP)
SEPPP = SEPP(IRSEP)
12220
CONTINUE
TEMGAT = TGATE(JTEMP) + SEPPP/SVEL(ITEMP)
CALL
CALL UPKUNI(ICLASS, IEIGHT, IPERFR7, 1., TEMPS)
TEMPS = TEMPS/3600./60.
DECCOR = 0.

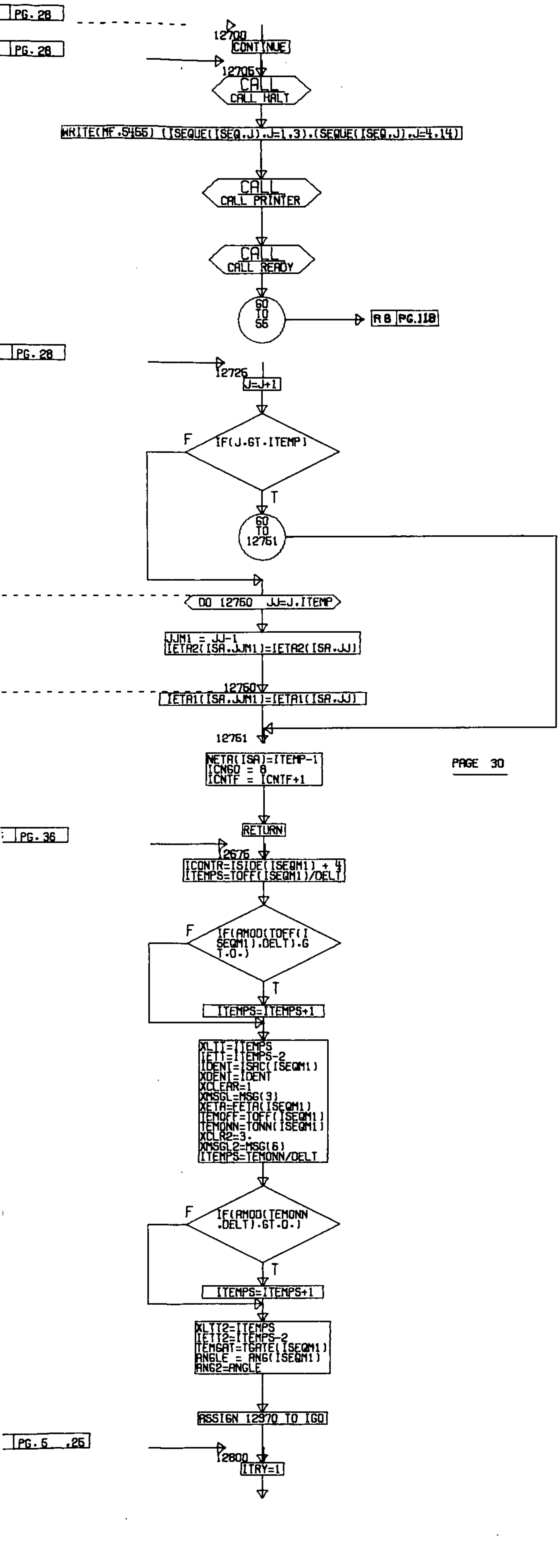


PAGE 21
IF(IHISIDE.EQ.1 .AND. INDR.EQ.0)
DECCOR = (ATRAF6(IAC)-SVEL(ITEMP))**2 / (2.0*TEMPS*SVEL(ITEMP))
IF(IHISIDE.EQ.1 .AND. INDR.EQ.0 .AND. TYI.LT.16.0)
DECCOR = DECCOR + 4.7/SVEL(ITEMP) - 4.7/ATRAF6(IAC)
TEMGAT = TEMGAT + SEPCOR - DECCOR
A=RA20(ITEMP)
B=RB20(ITEMP)
IANG=1
R1=.5*((TEMGAT-TARR(ITEMP))*TEMP-A-B)
R1 = R1 + 0.16
IF(IHISIDE .EQ . 2)
R1 = R1+0.32
IF(SEPCOR.EQ.0. .AND. TARR(ITEMP).GT.TARR(JTEMP))
R1 = R1 + (1.-SVEL(ITEMP)/SVEL(JTEMP)) * (R1+XDIST(IHISIDE))
PAGE 22
IF(INDR.EQ.0 .AND. R1.LT.DL(ITEMP)-0.15-15.0*SVEL(ITEMP))
R1 = DL(ITEMP)-0.15-15.0*SVEL(ITEMP)
IF(INDR.EQ.1 .AND. R1.LT.DL(ITEMP))
R1 = DL(ITEMP)
PG.18 .19
12230
IF(R1.LE.(2.-XDIST(IHEM)))
GO TO 12275
B11 PG.23
A=RA30(ITEMP)
B=RB30(ITEMP)
IANG=2
IF(ISEQ.LT.3)
GO TO 12275
B11 PG.23
PAGE 23
R1=.5*((TEMGAT-TARR(ITEMP))*TEMP-A-B)
R1 = R1 + 0.16
IF(IHISIDE .EQ . 2)
R1 = R1+0.32
IF(SEPCOR.EQ.0. .AND. TARR(ITEMP).GT.TARR(JTEMP))
R1 = R1 + (1.-SVEL(ITEMP)/SVEL(JTEMP)) * (R1+XDIST(IHISIDE))
IF(INDR.EQ.0 .AND. R1.LT.DL(ITEMP)-0.15-15.0*SVEL(ITEMP))
R1 = DL(ITEMP)-0.15-15.0*SVEL(ITEMP)
IF(INDR.EQ.1 .AND. R1.LT.DL(ITEMP))
R1 = DL(ITEMP)
B11 PG.22 .22
12275
PAGE 24
TEMOFF=TARR(ITEMP)+R1/TEMP
TEMONN = TEMOFF+A/TEMP
TEMGAT = TEMONN+(R1+B)/TEMP
12300
IF(IND-1) 12350,12326,
B15 PG.25
12326
TGATE(ISEQ) = TEMGAT
TOFF(ISEQ) = TEMOFF
TONN(ISEQ) = TEMONN
JOE = ISIDE(ISEQ)
ANG(ISEQ)=ITANG(JOE ,IANG)
IND= 2
ITEMP = ISEQM1
JTEMP = ISEQ
GO TO 12200
B8 PG.19
12350
TOFF(ISEQ) = TEMOFF
TONN(ISEQ) = TEMONN
TGATE(ISEQ) = TEMGAT
JOE = ISIDE(ISEQ)
ANG(ISEQ)=ITANG(JOE ,IANG)
ANGLE=ANG(ISEQ)
ANG2 = ANGLE
ICONTR=IHEM+4
ITEMPS=TEMOFF/DELT
IF(AMOD(TEMOFF,DELT).GT.0.)
ITEMPS=ITEMPS+1
IFTI=ITEMPS-2
XLTT=ITEMPS
IDENT=ISAC(ISEQ)
XDENT=IDENT
ITEMPS=TEMONN/DELT

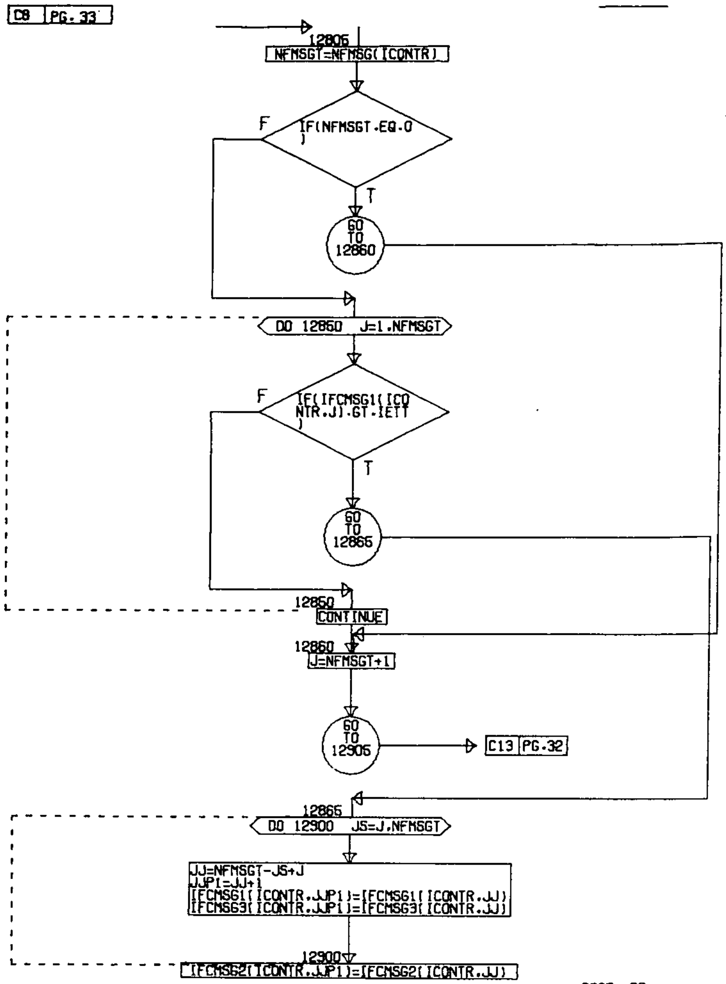

IF(AMOD(TEMONN .DELT).GT.0.)
F
T
ITEMPS=ITEMPS+1
IETT2=ITEMPS-2
XLTI2=ITEMPS
XCLR2=3
XMSGL2=MSG(6)
XCLEAR=1.
XMSGL=MSG(3)
XETA=FETA(ISEQ)
ASSIGN 12970 TO IGO
GO TO 12800
C7 PG.30
B16 PG.24
C**** UPDATE FUTURE MESSAGE ARRAY
12376
IF(TEMOFF.LE.T CP(ISEQM1))
F
T
GO TO 12400
IND = 0
ITEMP = ISEQ
JTEMP = ISEQM1
GO TO 12200
B8 PG.19
C**** EXCHANGE A/C AND PRECEDING A/C
12400
DO 12500 K=1,10
B17 PG.26
PAGE 26
TEMP= SEQUE(ISEQ,K)
SEQUE(ISEQ,K) =SEQUE(ISEQM1,K)
B17 PG.26
12500
SEQUE(ISEQM1,K) = TEMP
ANGLE=ANG(ISEQM1)
ANG(ISEQM1)=ANG(ISEQ)
ANG(ISEQ)=ANGLE
TOFF(ISEQM1)=TOFF(ISEQ)
TOFF(ISEQ) = TEMOFF
TONN(ISEQM1) = TONN(ISEQ)
TONN(ISEQ) = TEMONN
TGATE(ISEQM1) = TGATE(ISEQ)
TGATE(ISEQ) = TEMGAT
TEMP=DL(ISEQ)
DL(ISEQ)=DL(ISEQM1)
DL(ISEQM1)=TEMP
TEMP=FETA(ISEQ)
FETA(ISEQ)=FETA(ISEQM1)
FETA(ISEQM1)=TEMP
C**** UPDATE FUTURE MESSAGE ARRAY
ICONTR=ISIDE(ISEQ)+4
ITEMPS=TEMOFF/DELT
IF(AMOD(TEMOFF .DELT).GT.0.)
F
T
ITEMPS=ITEMPS+1
IETT=ITEMPS-2
XLTT=ITEMPS
ITEMPS=TEMONN/DELT
IF(AMOD(TEMOFF .DELT).GT.0.)
F
T
ITEMPS=ITEMPS+1


PAGE
IETT2=ITEMPS-2
XLTI2=ITEMPS
ANG2=ANGLE
XCLR2=3.
XMSGL2=MSG(6)
IDENT=ISAC(ISEQ)
XDENT=IDENT
XCLEAR=1.
XMSGL=MSG(3)
XETA=FETA(ISEQ)
NFMSGT=NFMSG(ICONTR)
IF(NFMSGT.LE.0)
F
T
GO TO 12606
B19 PG.28
K=0
DO 12600 J=1,NFMSGT
B18 PG.28
CALL
CALL UPKSPL(IONE,IONE,IFCMSG2(ICONTR,J),IPMSG2,TEMPS)
CALL
CALL UPKSPL(ISIX,ISIX,IFCMSG2(ICONTR,J),IPMSG2,TEMP)
IF(TEMPS.EQ.XDE NT.AND.TEMP.EQ. FETA(ISEQ))
F
T
GO TO 12600
B18 PG.28
K=K+1
IFCMSG1(ICONTR,K)=IFCMSG1(ICONTR,J)
IFCMSG2(ICONTR,K)=IFCMSG2(ICONTR,J)
IFCMSG3(ICONTR,K)=IFCMSG3(ICONTR,J)
PAGE
B18 PG.27 ,27
12600
CONTINUE
NFMSG(ICONTR)=K
B19 PG.27
12605
CONTINUE
ASSIGN 12675 TO IGO
ISA=FETA(ISEQ)
12685
ITEMP=NETA(ISA)
IF(ITEMP.EQ.0)
F
T
GO TO 12706
C2 PG.29
DO 12700 J=1,ITEMP
C1 PG.
CALL
CALL UPKSPL(IONE,IONE,IETA1(ISA,J),IPETA1,TEMPS)
ITEMPS=TEMPS
IF(ITEMPS.EQ.I DENT)
F
T
GO TO 12726
C3 PG.29

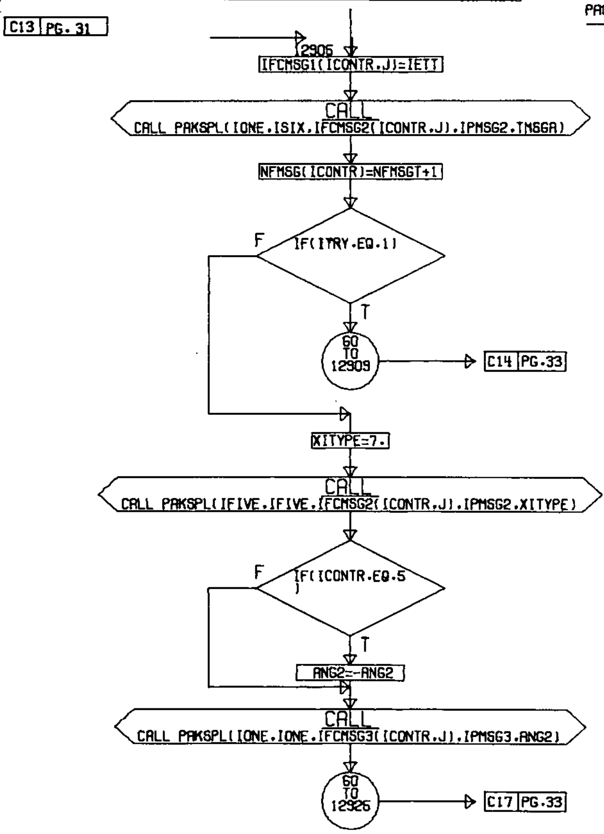
PAGE 29

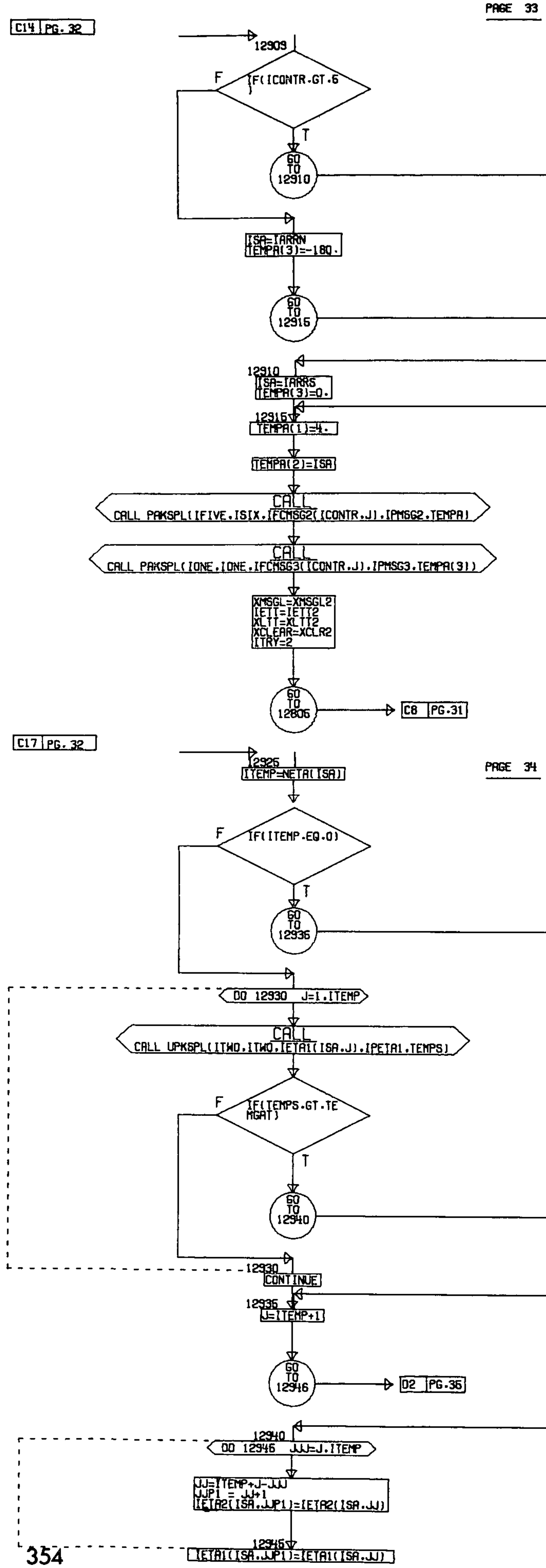


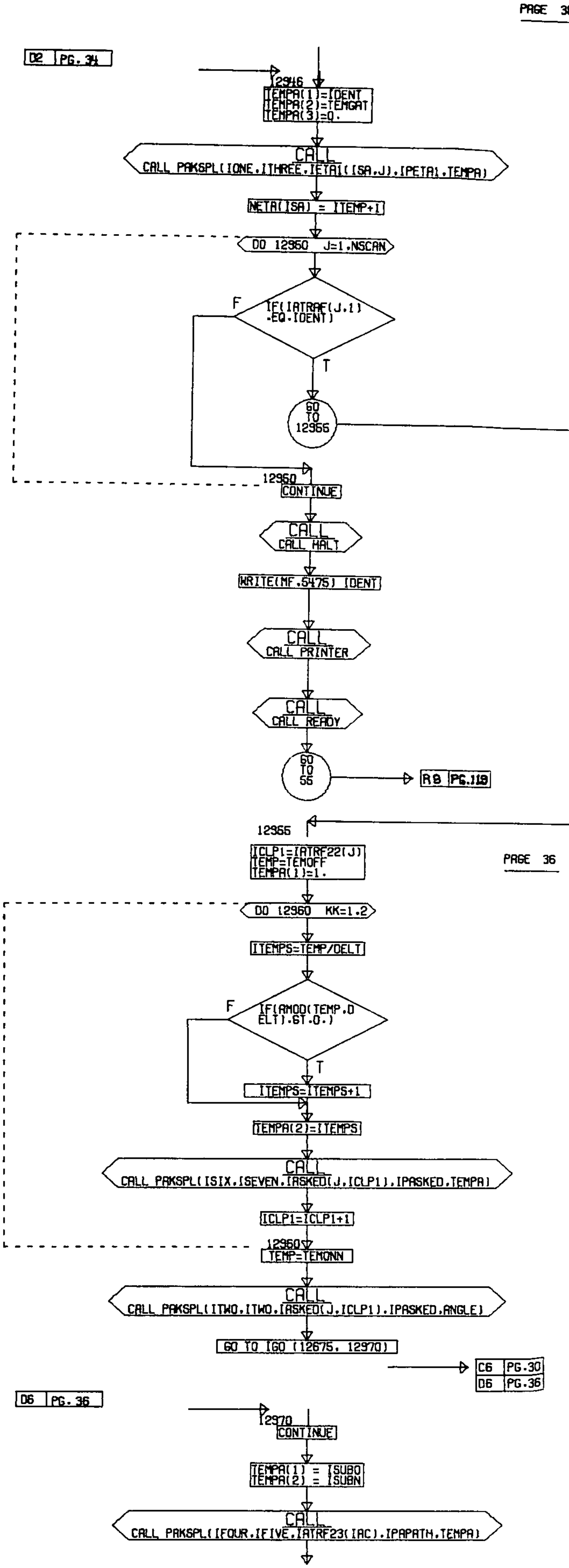
PAGE 31



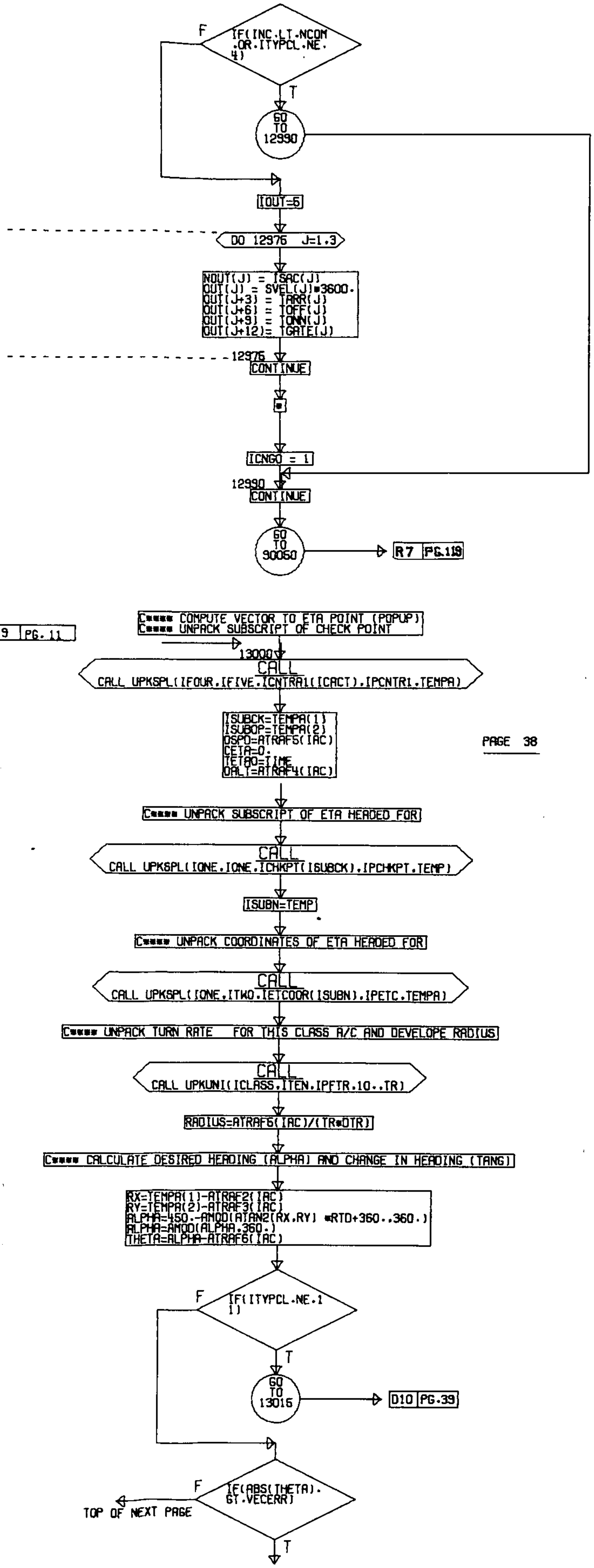
PAGE 32




PAGE 33
C14 PG. 32
12909
IF(ICONTR.GT.6)
F
T
GO TO 12910
ISA=IARRN
TEMPA(3)=-180.
GO TO 12915
12910
ISA=IARRS
TEMPA(3)=0.
12915
TEMPA(1)=4.
TEMPA(2)=ISA
CALL
CALL PAKSPL(IFIVE.ISIX.IFCMSG2(ICONTR.J).IPMSG2.TEMPA)
CALL
CALL PAKSPL(IONE.IONE.IFCMSG3(ICONTR.J).IPMSG3.TEMPA(3))
XMSGL=XMSGL2
IETI=IETI2
XLTI=XLTI2
XCLEAR=XCLR2
ITRY=2
GO TO 12806
C8 PG.31
C17 PG. 32
12925
ITEMP=NETA(ISA)
PAGE 34
IF(ITEMP.EQ.0)
F
T
GO TO 12936
DO 12930 J=1.ITEMP
CALL
CALL UPKSPL(ITWO.ITWO.IETA1(ISA.J).IPETA1.TEMPS)
IF(TEMPS.GT.TEMGAT)
F
T
GO TO 12940
12930
CONTINUE
12936
J=ITEMP+1
GO TO 12946
D2 PG.35
12940
DO 12946 JJJ=J.ITEMP
JJ=ITEMP+J-JJJ
JJP1 = JJ+1
IETA2(ISA.JJP1)=IETA2(ISA.JJ)
12946
IETA1(ISA.JJP1)=IETA1(ISA.JJ)

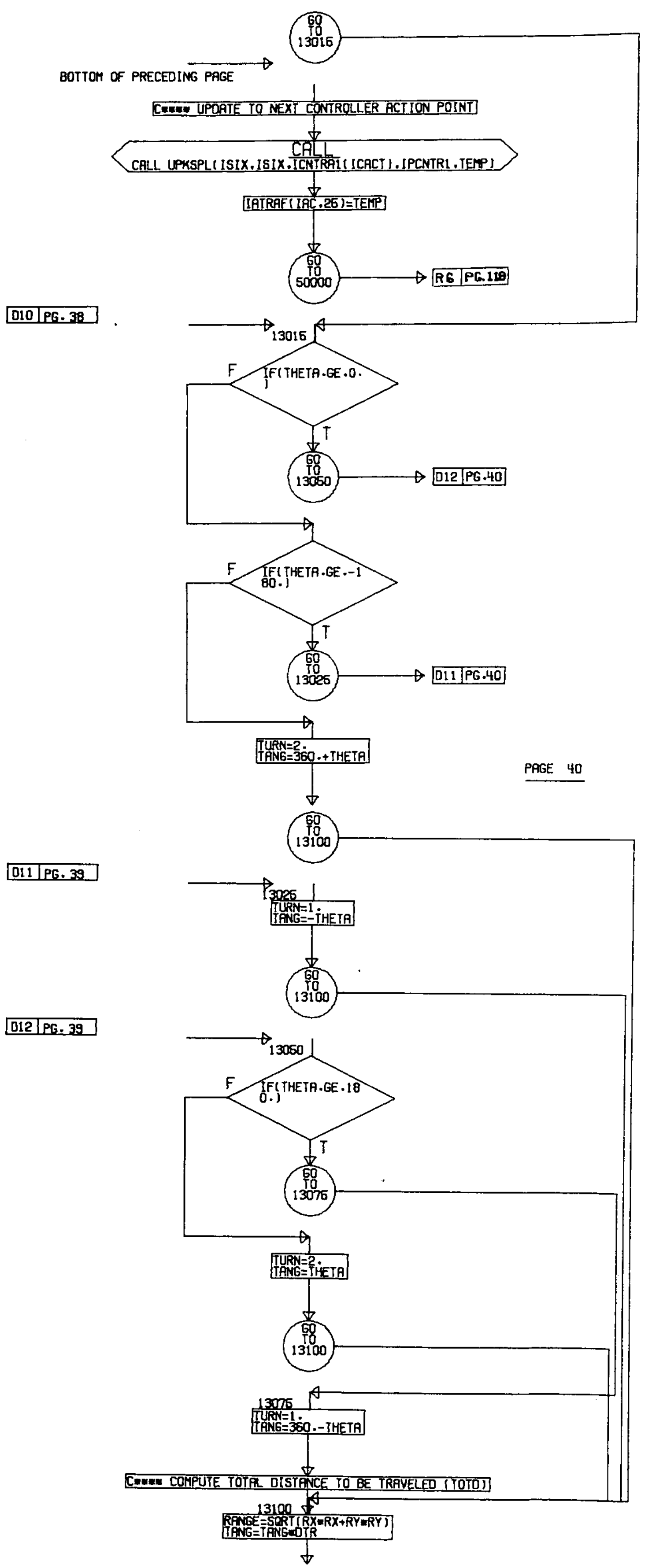

PAGE 35
D2 PG. 34
12946
TEMPA(1)=IDENT
TEMPA(2)=TEMGAT
TEMPA(3)=0.
CALL
CALL PAKSPL(IONE.ITHREE.IETA1(ISA.J).IPETA1.TEMPA)
NETA(ISA) = ITEMP+1
DO 12960 J=1.NSCAN
IF(IATRAF(J.1).EQ.IDENT)
F
T
GO TO 12965
12960
CONTINUE
CALL
CALL HALT
WRITE(MF.5475) IDENT
CALL
CALL PRINTER
CALL
CALL READY
GO TO 55
R9 PG.119
12965
ICLP1=IATRF22(J)
TEMP=TEMOFF
TEMPA(1)=1.
PAGE 36
DO 12960 KK=1.2
ITEMPS=TEMP/DELT
IF(AMOD(TEMP.DELT).GT.0.)
F
T
ITEMPS=ITEMPS+1
TEMPA(2)=ITEMPS
CALL
CALL PAKSPL(ISIX.ISEVEN.IASKED(J.ICLP1).IPASKED.TEMPA)
ICLP1=ICLP1+1
12960
TEMP=TEMONN
CALL
CALL PAKSPL(ITWO.ITWO.IASKED(J.ICLP1).IPASKED.ANGLE)
GO TO IGO (12675. 12970)
C6 PG.30
D6 PG.36
D6 PG. 36
12970
CONTINUE
TEMPA(1) = ISUBO
TEMPA(2) = ISUBN
CALL
CALL PAKSPL(IFOUR.IFIVE.IATRF23(IAC).IPAPATH.TEMPA)

PAGE 37

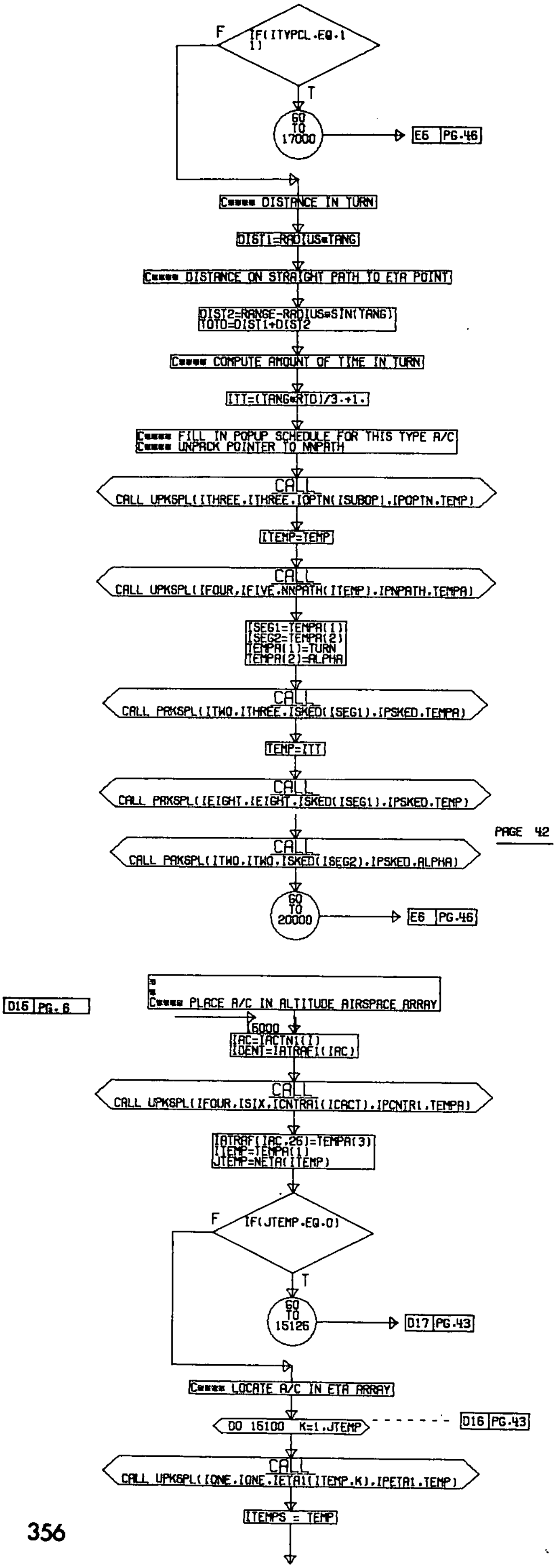


PAGE 38

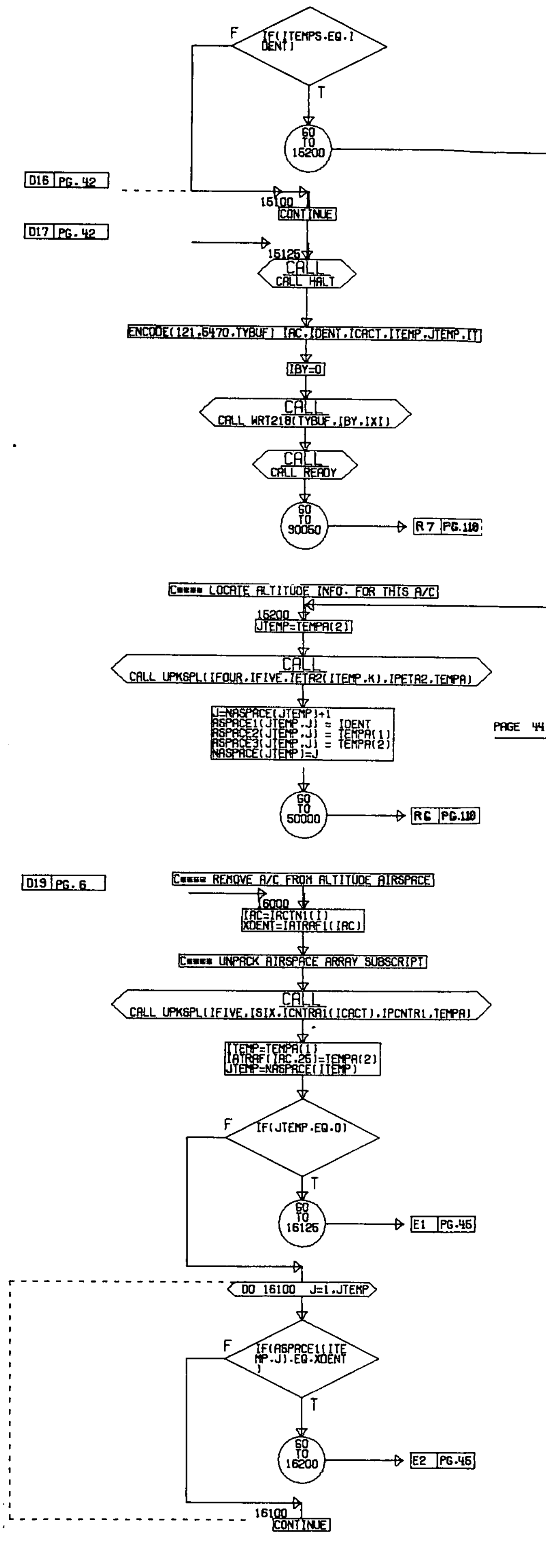
PAGE 39



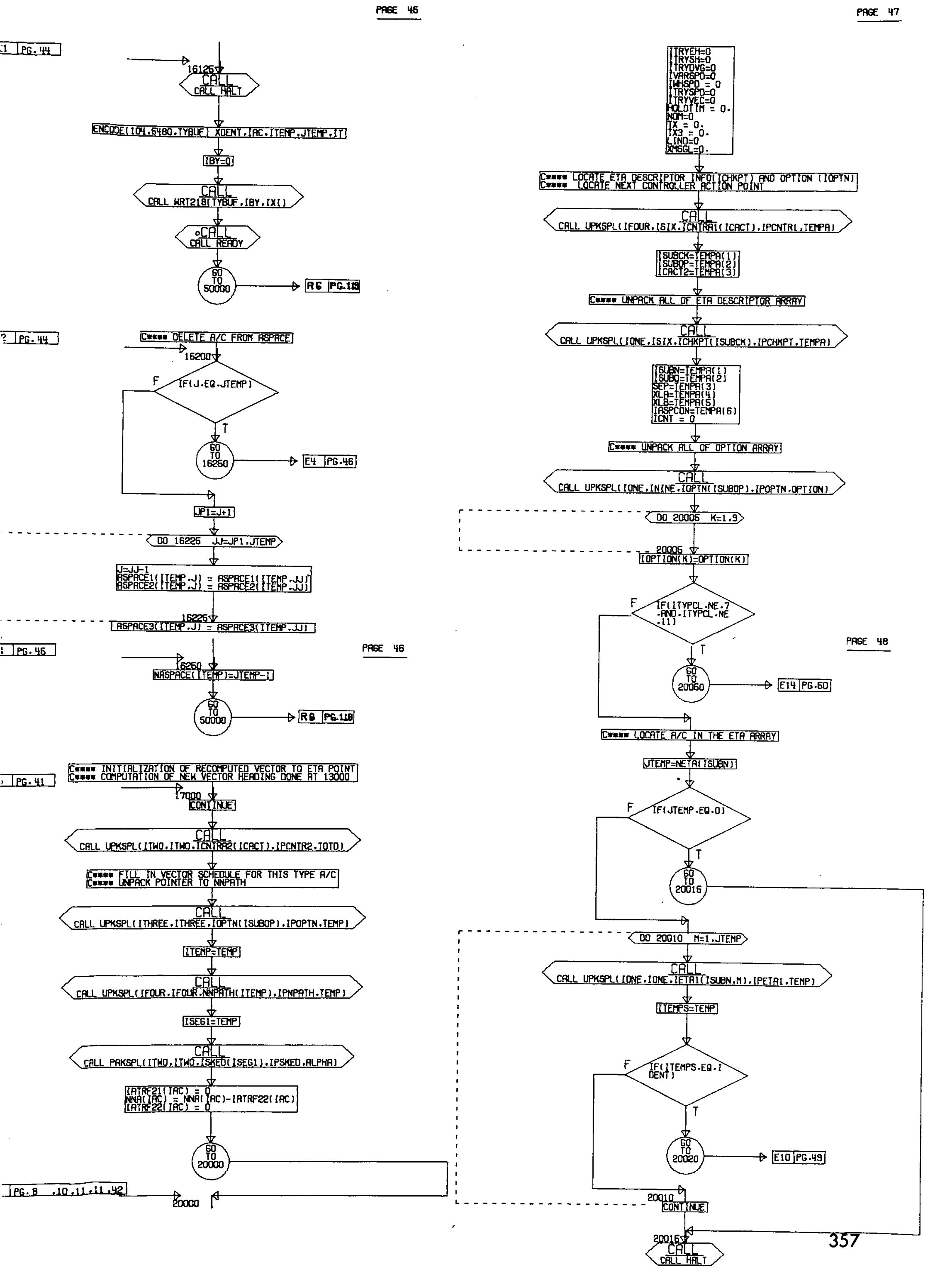
PAGE 40

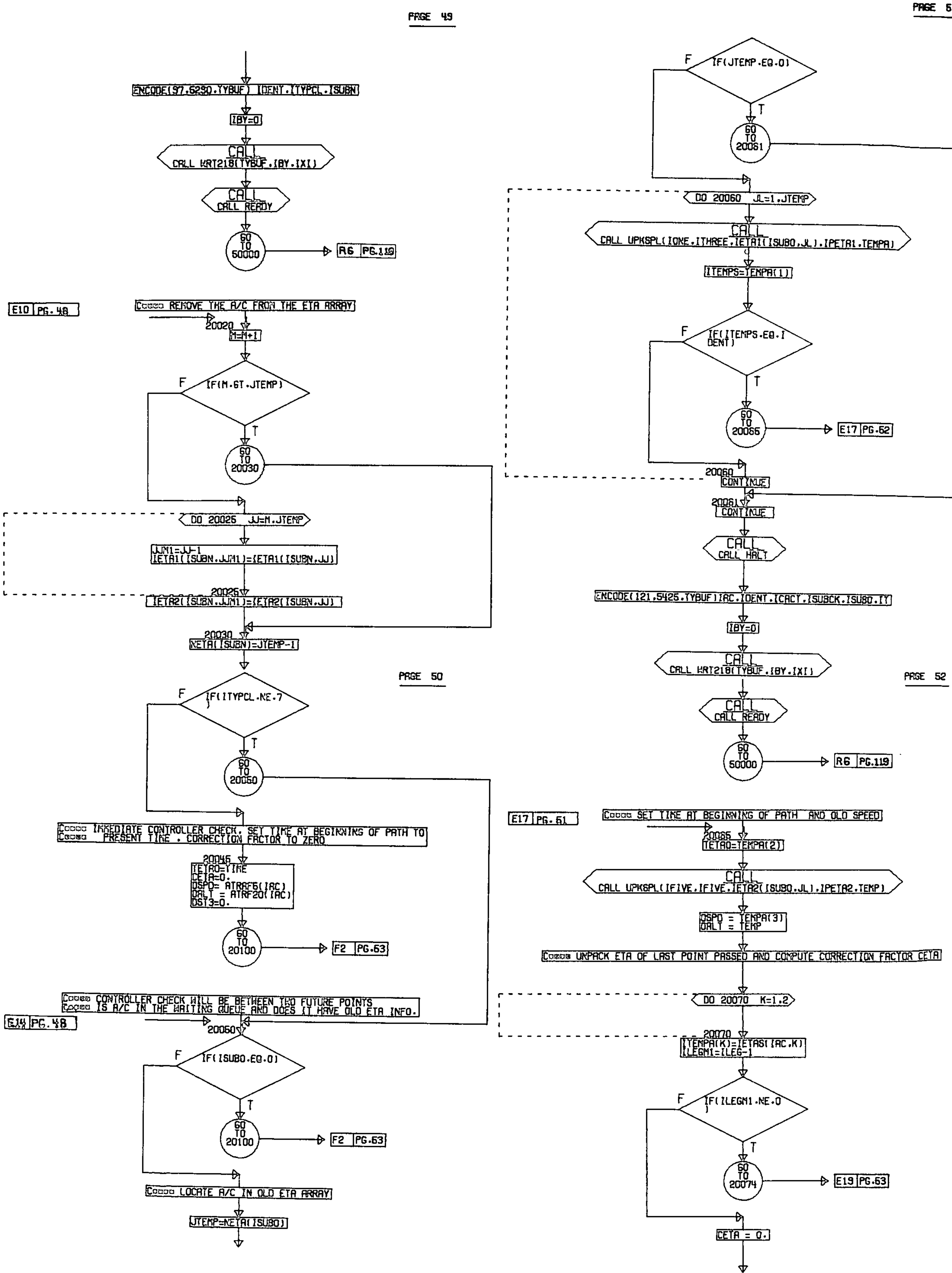


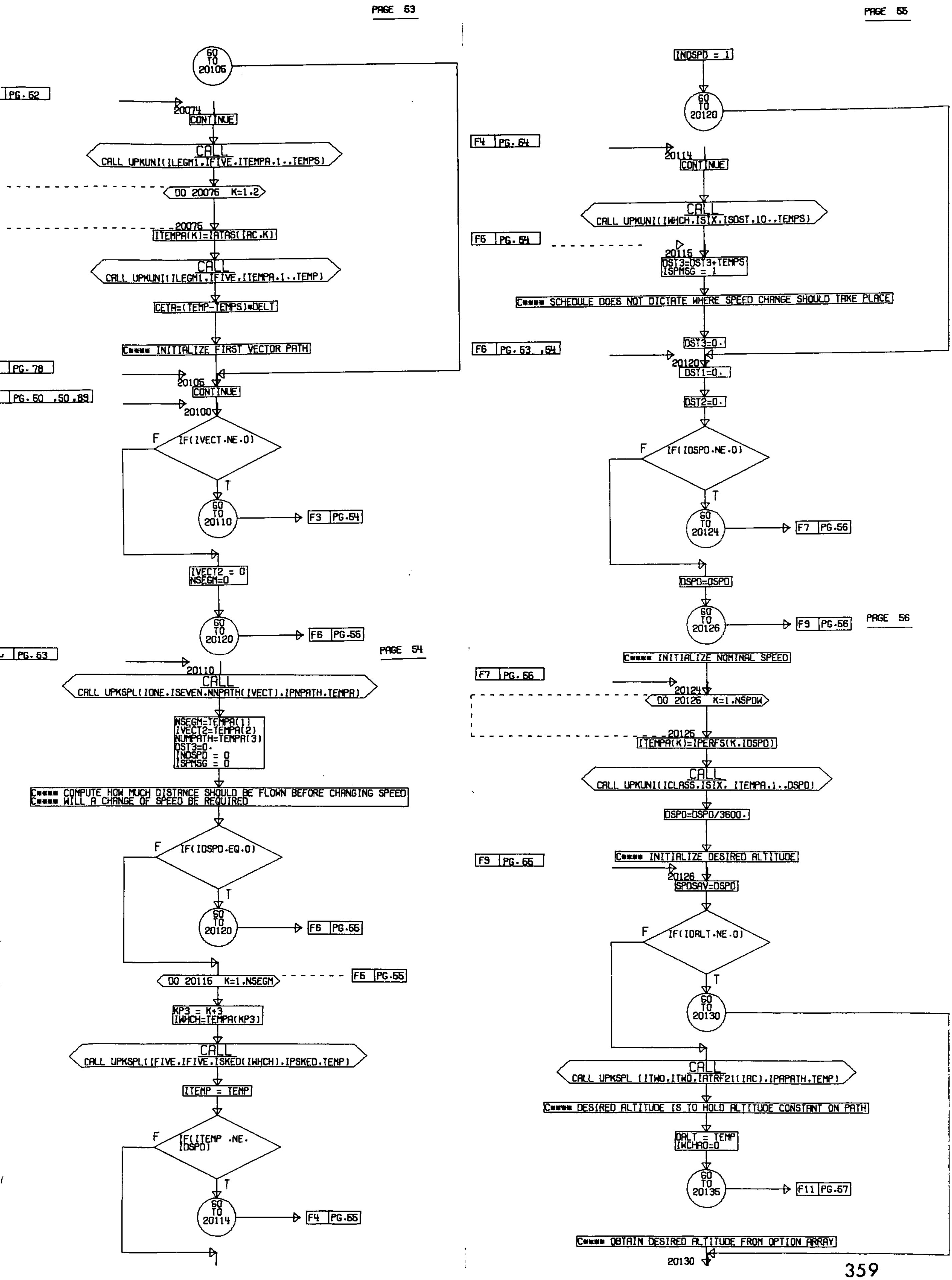
F
IF(ITYPCL.EQ.1 1)
T
GO TO 17000
E6 PG.46
C***** DISTANCE IN TURN
DIST1=RADIUS*TANG
C***** DISTANCE ON STRAIGHT PATH TO ETA POINT
DIST2=RANGE-RADIUS*SIN(TANG)
TOTD=DIST1+DIST2
C***** COMPUTE AMOUNT OF TIME IN TURN
ITT=(TANG*RTD)/3.+1.
C***** FILL IN POPUP SCHEDULE FOR THIS TYPE A/C
C***** UNPACK POINTER TO NNPATH
CALL
CALL UPKSPL(ITHREE,ITHREE,IOPTN(ISUBOP),IPOPTN,TEMP)
ITEMP=TEMP
CALL
CALL UPKSPL(IFOUR,IFIVE,NNPATH(ITEMP),IPNPATH,TEMPA)
ISEG1=TEMPA(1)
ISEG2=TEMPA(2)
TEMPA(1)=TURN
TEMPA(2)=ALPHA
CALL
CALL PAKSPL(ITWO,ITHREE,ISKED(ISEG1),IPSKED,TEMPA)
TEMP=ITT
CALL
CALL PAKSPL(IEIGHT,IEIGHT,ISKED(ISEG1),IPSKED,TEMP)
PAGE 42
CALL
CALL PAKSPL(ITWO,ITWO,ISKED(ISEG2),IPSKED,ALPHA)
GO TO 20000
E6 PG.46
D16 PG.6
C
C
C***** PLACE A/C IN ALTITUDE AIRSPACE ARRAY
16000
IAC=IACTN1(I)
IDENT=IATRAF1(IAC)
CALL
CALL UPKSPL(IFOUR,ISIX,ICNTRA1(ICACT),IPCNTR1,TEMPA)
IATRAF(IAC,26)=TEMPA(3)
ITEMP=TEMPA(1)
JTEMP=NETA(ITEMP)
F
IF(JTEMP.EQ.0)
T
GO TO 15126
D17 PG.43
C***** LOCATE A/C IN ETA ARRAY
DO 15100 K=1,JTEMP
D16 PG.43
CALL
CALL UPKSPL(IONE,IONE,IETA1(ITEMP,K),IPETA1,TEMP)
ITEMPS = TEMP

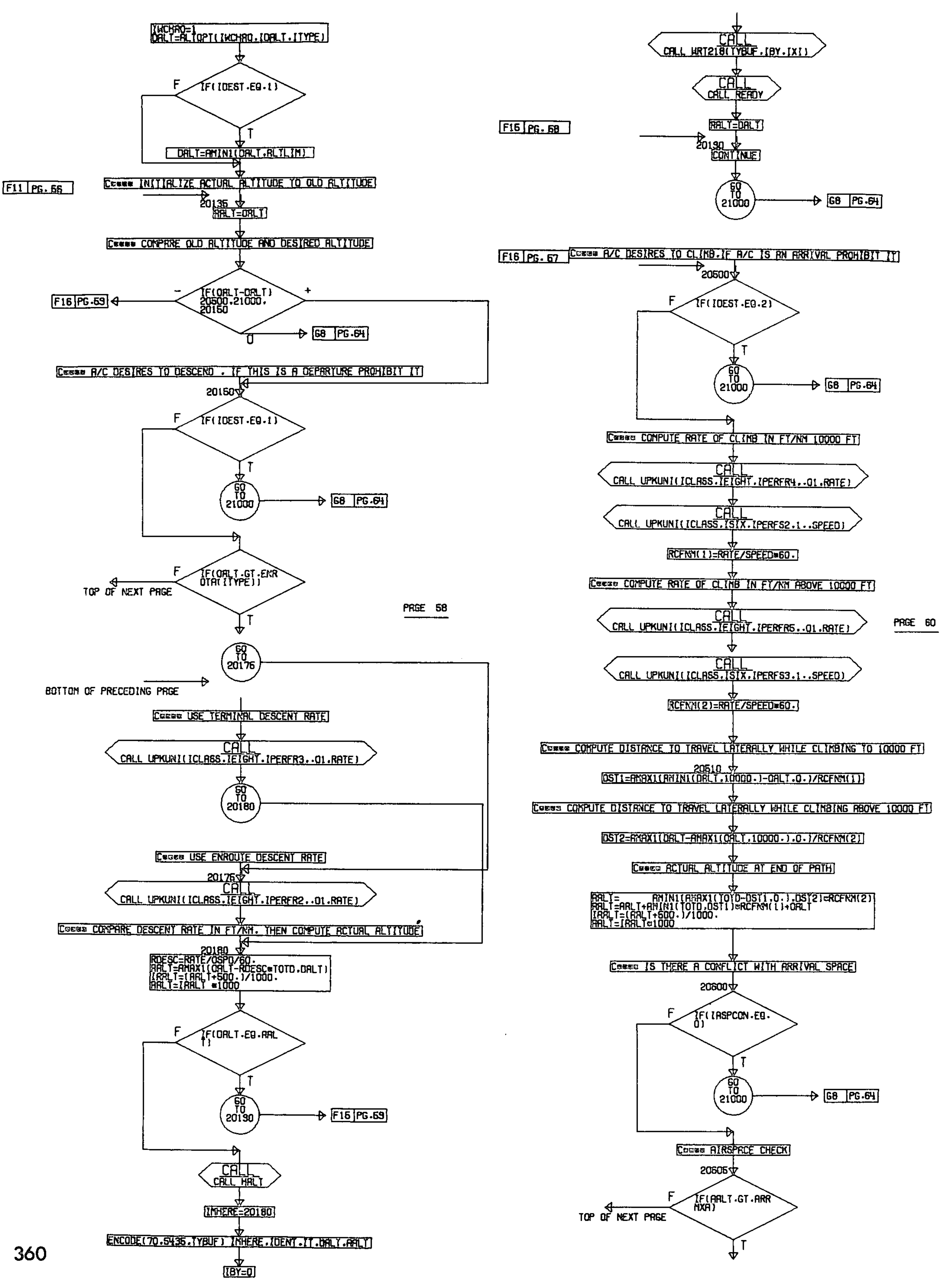


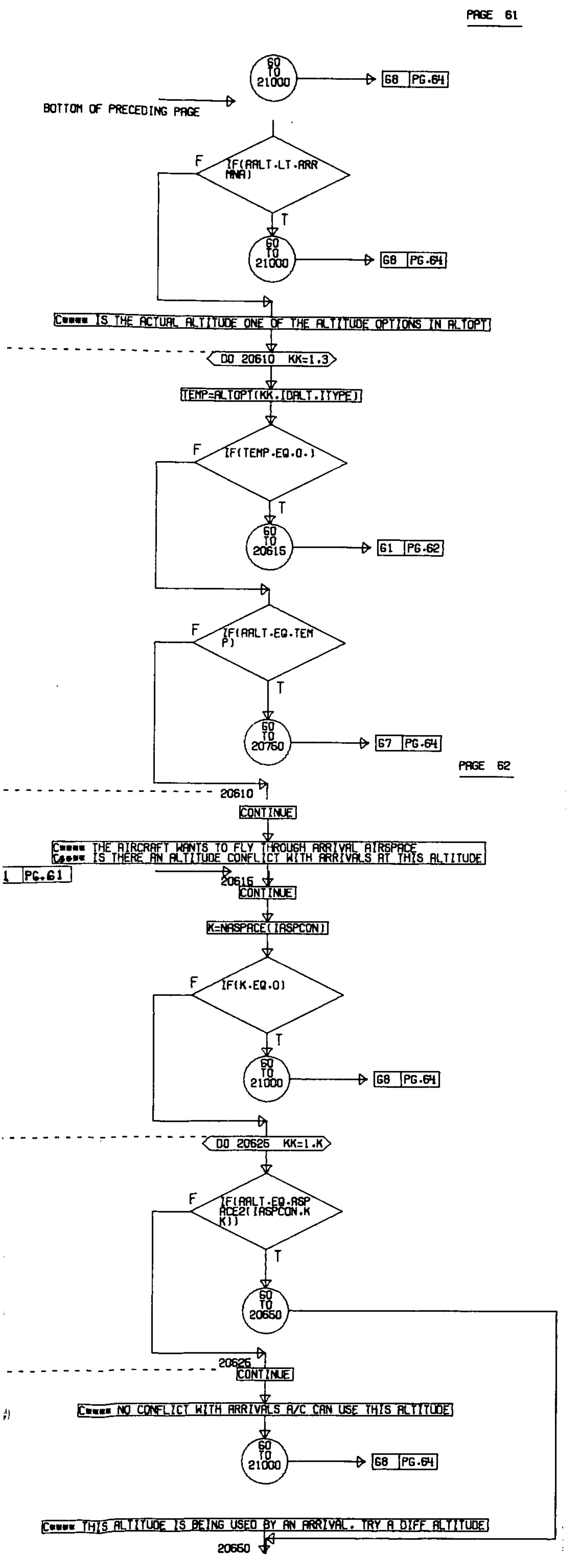
F
IF(ITEMPS.EQ.I DENT)
T
GO TO 15200
D16 PG.42
15100
CONTINUE
D17 PG.42
15125
CALL
CALL HALT
ENCODE(121,5470,TYBUF) IAC,IDENT,ICACT,ITEMP,JTEMP,ITT
IBY=0
CALL
CALL WRT2(8(TYBUF,IBY,IXI)
CALL
CALL READY
GO TO 30050
R7 PG.110
C***** LOCATE ALTITUDE INFO. FOR THIS A/C
15200
JTEMP=TEMPA(2)
CALL
CALL UPKSPL(IFOUR,IFIVE,IETA2(ITEMP,K),IPETA2,TEMPA)
J=NASPACE(JTEMP)+1
ASPACE1(JTEMP,J) = IDENT
ASPACE2(JTEMP,J) = TEMPA(1)
ASPACE3(JTEMP,J) = TEMPA(2)
NASPACE(JTEMP)=J
PAGE 44
GO TO 50000
R6 PG.110
D19 PG.6
C***** REMOVE A/C FROM ALTITUDE AIRSPACE
16000
IAC=IACTN1(I)
XDENT=IATRAF1(IAC)
C***** UNPACK AIRSPACE ARRAY SUBSCRIPT
CALL
CALL UPKSPL(IFIVE,ISIX,ICNTRA1(ICACT),IPCNTR1,TEMPA)
ITEMP=TEMPA(1)
IATRAF(IAC,26)=TEMPA(2)
JTEMP=NASPACE(ITEMP)
F
IF(JTEMP.EQ.0)
T
GO TO 16126
E1 PG.45
DO 16100 J=1,JTEMP
F
IF(ASPACE1(ITE MP,J).EQ.XDENT)
T
GO TO 16200
E2 PG.45
16100
CONTINUE

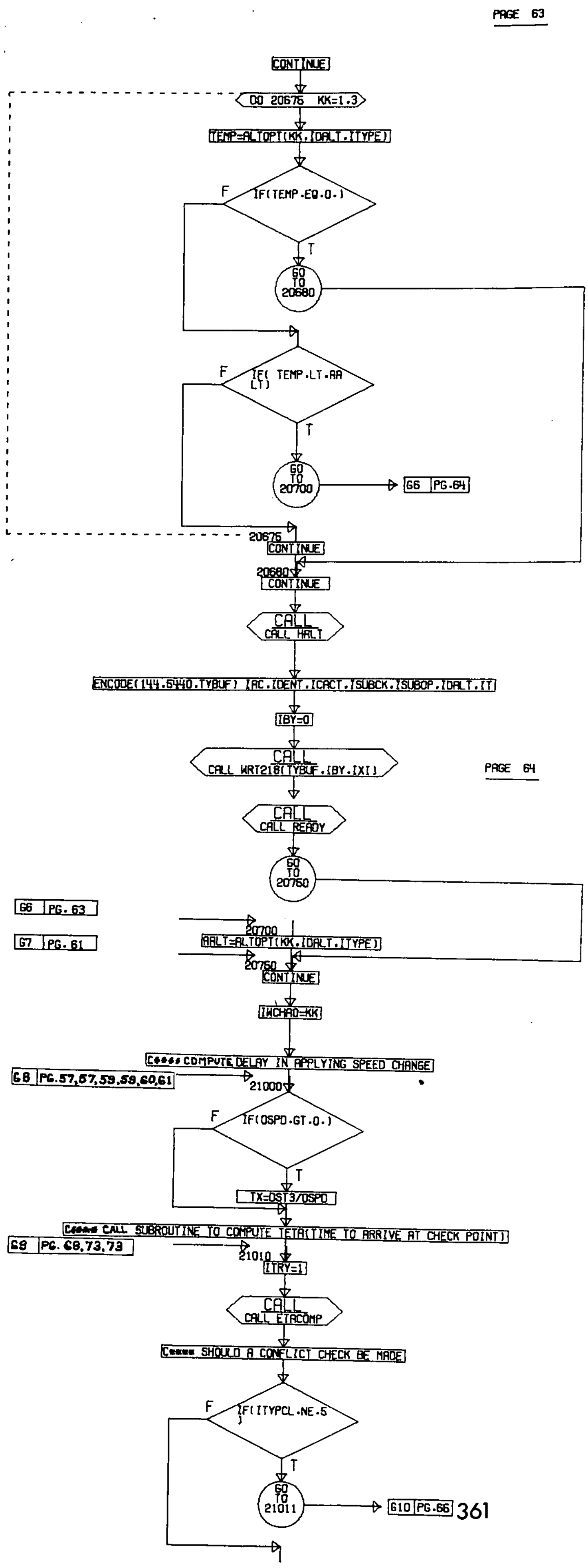

PAGE 46
PG. 44
16125
CALL
CALL HALT
ENCODE(104.6480.TYBUF) XDENT.IAC.ITEMP.JTEMP.IT)
IBY=0
CALL
CALL WRT218(TYBUF.IBY.IX1)
CALL
CALL READY
GO TO 50000
R6 PG.118
PG. 44
C***** DELETE A/C FROM ASPACE
16200
F
IF(J.EQ.JTEMP)
T
GO TO 16250
E4 PG.46
JP1=J+1
DO 16225 JJ=JP1.JTEMP
J=JJ-1
ASPACE1(ITEMP.J) = ASPACE1(ITEMP.JJ)
ASPACE2(ITEMP.J) = ASPACE2(ITEMP.JJ)
16225
ASPACE3(ITEMP.J) = ASPACE3(ITEMP.JJ)
PG. 46
PAGE 46
16250
NASPACE(ITEMP)=JTEMP-1
GO TO 50000
R6 PG.118
C***** INITIALIZATION OF RECOMPUTED VECTOR TO ETA POINT
C***** COMPUTATION OF NEW VECTOR HEADING DONE AT 13000
PG. 41
17000
CONTINUE
CALL
CALL UPKSPL(ITWO.ITWO.ICNTRA2(ICACT).IPCNTR2.TOTD)
C***** FILL IN VECTOR SCHEDULE FOR THIS TYPE A/C
C***** UNPACK POINTER TO NNPATH
CALL
CALL UPKSPL(ITHREE.ITHREE.IOPTN(ISUBOP).IPOPTN.TEMP)
ITEMP=TEMP
CALL
CALL UPKSPL(IFOUR.IFOUR.NNPATH(ITEMP).IPNPATH.TEMP)
ISEG1=TEMP
CALL
CALL PAKSPL(ITWO.ITWO.ISKED(ISEG1).IPSKED.ALPHA)
IATRF21(IAC) = 0
NNA(IAC) = NNA(IAC)-IATRF22(IAC)
IATRF22(IAC) = 0
GO TO 20000
PG. 8 .10.11.11.42
20000
PAGE 47
ITRYEH=0
ITRYSH=0
ITRYDVG=0
IVARSPD=0
IWHSPD = 0
ITRYSPD=0
ITRYVEC=0
HOLDTIM = 0.
NOM=0
TX = 0.
TX3 = 0.
LIND=0
XMSGL=0.
C***** LOCATE ETA DESCRIPTOR INFO(ICHKPT) AND OPTION (IOPTN)
C***** LOCATE NEXT CONTROLLER ACTION POINT
CALL
CALL UPKSPL(IFOUR.ISIX.ICNTRA1(ICACT).IPCNTR1.TEMPA)
ISUBCK=TEMPA(1)
ISUBOP=TEMPA(2)
ICACT2=TEMPA(3)
C***** UNPACK ALL OF ETA DESCRIPTOR ARRAY
CALL
CALL UPKSPL(IONE.ISIX.ICHKPT(ISUBCK).IPCHKPT.TEMPA)
ISUBN=TEMPA(1)
ISUBQ=TEMPA(2)
SEP=TEMPA(3)
XLA=TEMPA(4)
XLB=TEMPA(5)
IASPCON=TEMPA(6)
ICNT = 0
C***** UNPACK ALL OF OPTION ARRAY
CALL
CALL UPKSPL(IONE.ININE.IOPTN(ISUBOP).IPOPTN.OPTION)
DO 20006 K=1.9
20006
IOPTION(K)=OPTION(K)
F
IF(ITYPCL.NE.7.AND.ITYPCL.NE.11)
T
GO TO 20050
E14 PG.50
PAGE 48
C***** LOCATE A/C IN THE ETA ARRAY
JTEMP=NETA(ISUBN)
F
IF(JTEMP.EQ.0)
T
GO TO 20016
DO 20010 M=1.JTEMP
CALL
CALL UPKSPL(IONE.IONE.IETA1(ISUBN.M).IPETA1.TEMP)
ITEMPS=TEMP
F
IF(ITEMPS.EQ.IDENT)
T
GO TO 20020
E10 PG.49
20010
CONTINUE
20016
CALL
CALL HALT

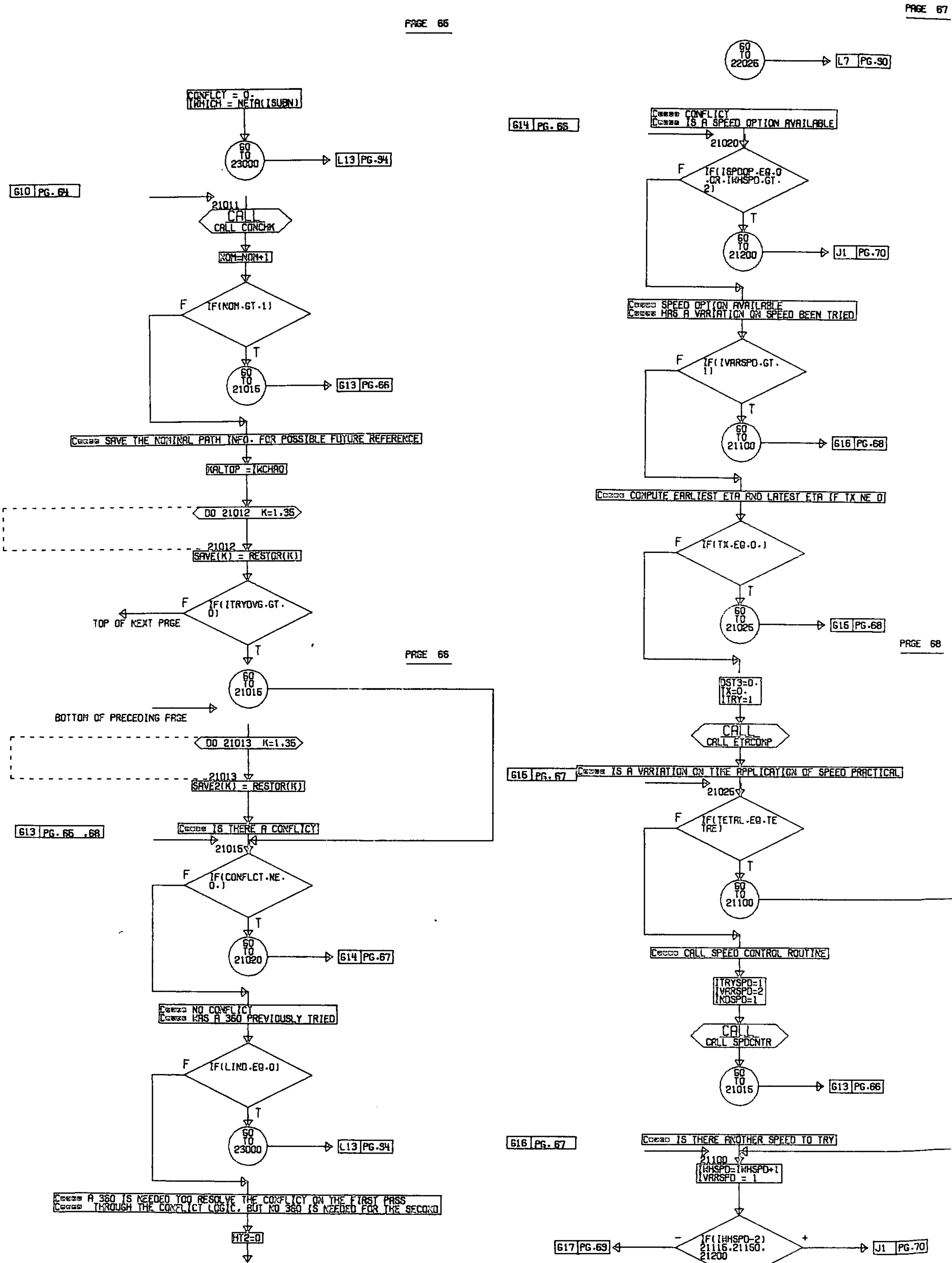

PAGE 49
ENCODE(97,5290,TYBUF) IDENT,ITYPCL,ISUBN
IBY=0
CALL
CALL WRT218(TYBUF,IBY,IXI)
CALL
CALL READY
GO TO 50000
R6 PG.119
E10 PG.48
Caaaa REMOVE THE A/C FROM THE ETA ARRAY
20020
M=M+1
IF(M.GT.JTEMP)
F
T
GO TO 20030
DO 20025 JJ=M,JTEMP
JJM1=JJ-1
IETA1(ISUBN,JJM1)=IETA1(ISUBN,JJ)
20025
IETA2(ISUBN,JJM1)=IETA2(ISUBN,JJ)
20030
NETA(ISUBN)=JTEMP-1
PAGE 50
IF(ITYPCL.NE.7)
GO TO 20050
Caaaa IMMEDIATE CONTROLLER CHECK. SET TIME AT BEGINNING OF PATH TO
Caaaa PRESENT TIME , CORRECTION FACTOR TO ZERO
20046
IETA0=TIME
CETA=0.
OSPD= ATRAF6(IAC)
OALT = ATRF20(IAC)
DST3=0.
GO TO 20100
F2 PG.63
Caaaa CONTROLLER CHECK WILL BE BETWEEN TWO FUTURE POINTS
Caaaa IS A/C IN THE WAITING QUEUE AND DOES IT HAVE OLD ETA INFO.
E14 PG.48
20050
IF(ISUB0.EQ.0)
GO TO 20100
F2 PG.63
Caaaa LOCATE A/C IN OLD ETA ARRAY
JTEMP=NETA(ISUB0)
PAGE 5
IF(JTEMP.EQ.0)
GO TO 20061
DO 20060 JL=1,JTEMP
CALL
CALL UPKSPL(IONE,ITHREE,IETA1(ISUB0,JL),IPETA1,TEMPA)
ITEMPS=TEMPA(1)
IF(ITEMPS.EQ.IDENT)
GO TO 20065
E17 PG.52
20060
CONTINUE
20061
CONTINUE
CALL
CALL HALT
ENCODE(121,5425,TYBUF)IAC,IDENT,ICACT,ISUBCK,ISUB0,IT
IBY=0
CALL
CALL WRT218(TYBUF,IBY,IXI)
CALL
CALL READY
GO TO 50000
R6 PG.119
PAGE 52
E17 PG.51
Caaaa SET TIME AT BEGINNING OF PATH AND OLD SPEED
20065
IETA0=TEMPA(2)
CALL
CALL UPK6PL(IFIVE,IFIVE,IETA2(ISUB0,JL),IPETA2,TEMP)
OSPD = TEMPA(3)
OALT = TEMP
Caaaa UNPACK ETA OF LAST POINT PASSED AND COMPUTE CORRECTION FACTOR CETA
DO 20070 K=1,2
20070
ITEMPA(K)=IETAS(IAC,K)
ILEGM1=ILEG-1
IF(ILEGM1.NE.0)
GO TO 20074
E19 PG.63
CETA = 0.

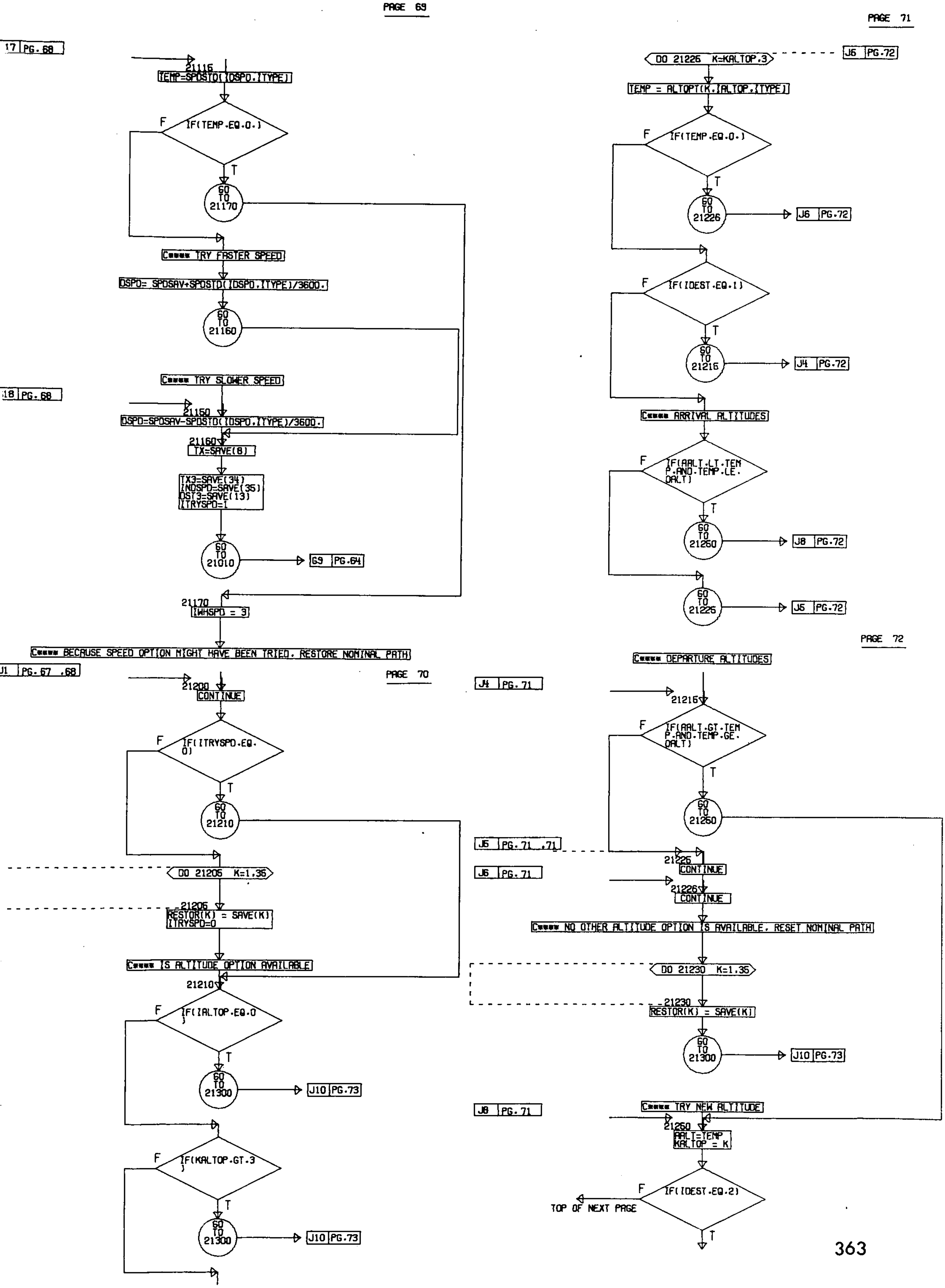

PAGE 53
GO TO 20106
PG. 52
20074 CONTINUE
CALL
CALL UPKUNI(ILEGM1,IFIVE,ITEMPA,1,,TEMPS)
DO 20076 K=1,2
20076 ITEMPA(K)=IATAS(IAC,K)
CALL
CALL UPKUNI(ILEGM1,IFIVE,ITEMPA,1,,TEMP)
CETA=(TEMP-TEMPS)*DELT
C**** INITIALIZE FIRST VECTOR PATH
PG. 78
20106 CONTINUE
PG. 50 ,50 ,89
20100
IF(IVECT.NE.0)
F
T
GO TO 20110
F3 PG.54
IVECT2 = 0
NSEGM=0
GO TO 20120
F6 PG.55
PG. 53
PAGE 54
20110
CALL
CALL UPKSPL(IONE,ISEVEN,NNPATH(IVECT),IPNPATH,TEMPA)
NSEGM=TEMPA(1)
IVECT2=TEMPA(2)
NUMPATH=TEMPA(3)
DST3=0.
INDSPD = 0
ISPMSG = 0
C**** COMPUTE HOW MUCH DISTANCE SHOULD BE FLOWN BEFORE CHANGING SPEED
C**** WILL A CHANGE OF SPEED BE REQUIRED
IF(IDSPD.EQ.0)
F
T
GO TO 20120
F6 PG.55
DO 20116 K=1,NSEGM
F5 PG.55
KP3 = K+3
IWHCH=TEMPA(KP3)
CALL
CALL UPKSPL(IFIVE,IFIVE,ISKED(IWHCH),IPSKED,TEMP)
ITEMP = TEMP
IF(ITEMP .NE. IDSPD)
F
T
GO TO 20114
F4 PG.55
PAGE 55
INDSPD = 1
GO TO 20120
F4 PG. 54
20114 CONTINUE
CALL
CALL UPKUNI(IWHCH,ISIX,ISDST,10,,TEMPS)
F5 PG. 54
20115
DST3=DST3+TEMPS
ISPMSG = 1
C**** SCHEDULE DOES NOT DICTATE WHERE SPEED CHANGE SHOULD TAKE PLACE
DST3=0.
F6 PG. 53 ,54
20120
DST1=0.
DST2=0.
IF(IOSPD.NE.0)
F
T
GO TO 20124
F7 PG.56
DSPD=OSPD
GO TO 20126
F9 PG.56
PAGE 56
C**** INITIALIZE NOMINAL SPEED
F7 PG. 55
20124
DO 20125 K=1,NSPDW
20125
ITEMPA(K)=IPERFS(K,IOSPD)
CALL
CALL UPKUNI(ICLASS,ISIX, ITEMPA,1,,DSPD)
DSPD=DSPD/3600.
C**** INITIALIZE DESIRED ALTITUDE
F9 PG. 55
20126
SPDSAV=DSPD
IF(IOALT.NE.0)
F
T
GO TO 20130
CALL
CALL UPKSPL (ITWO,ITWO,IATRF21(IAC),IPAPATH,TEMP)
C**** DESIRED ALTITUDE IS TO HOLD ALTITUDE CONSTANT ON PATH
DALT = TEMP
IWCHAO=0
GO TO 20135
F11 PG.57
C**** OBTAIN DESIRED ALTITUDE FROM OPTION ARRAY
20130

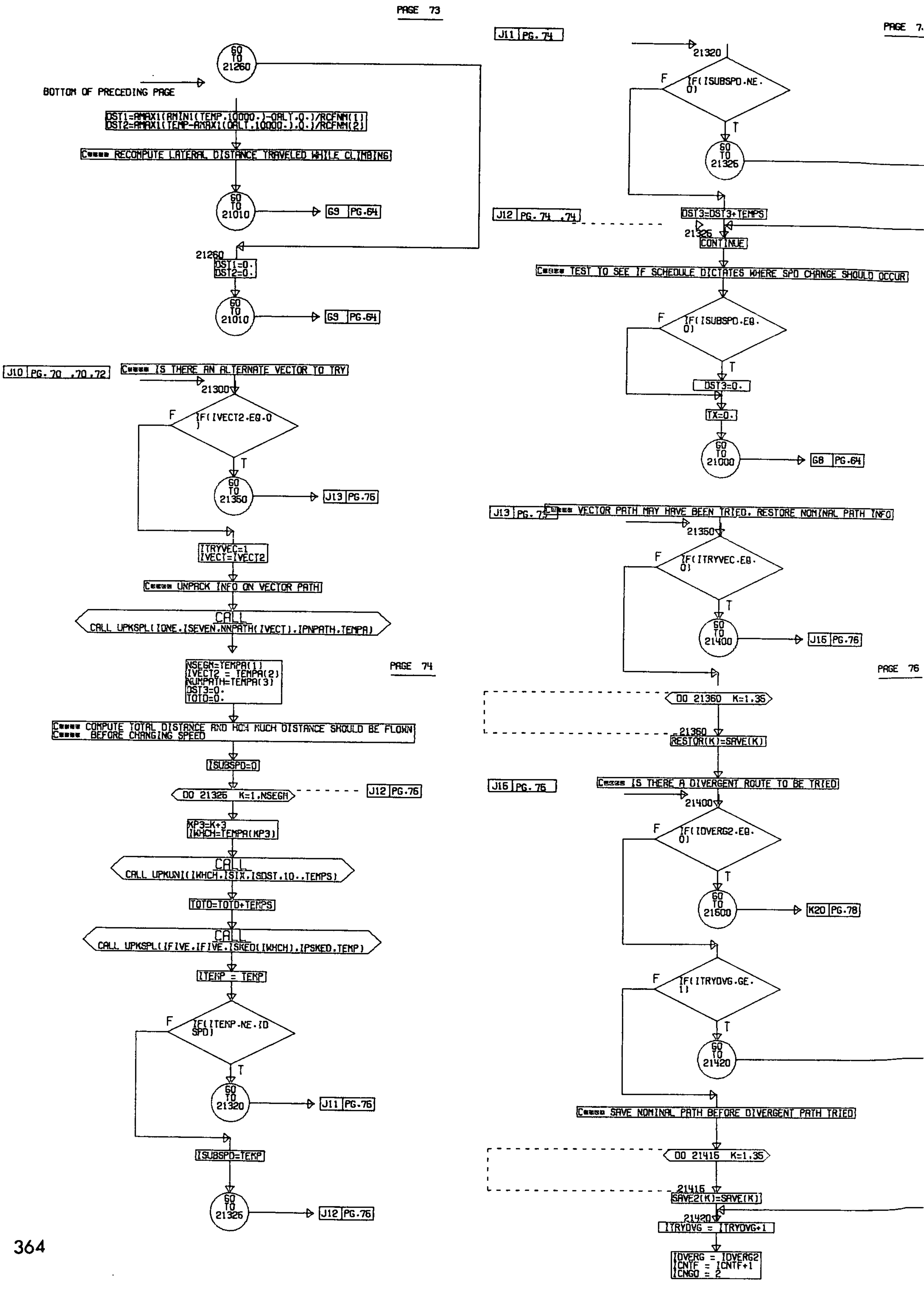

IWCHAQ=1
DALT=ALTOPT(IWCHAQ,IDALT,ITYPE)
IF(IDEST.EQ.1)
F
T
DALT=AMIN1(DALT,ALTLIM)
F11 PG.56
C**** INITIALIZE ACTUAL ALTITUDE TO OLD ALTITUDE
20135
AALT=OALT
C**** COMPARE OLD ALTITUDE AND DESIRED ALTITUDE
IF(OALT-DALT) 20500,21000,20150
F16 PG.59
-
+
0
G8 PG.64
C**** A/C DESIRES TO DESCEND . IF THIS IS A DEPARTURE PROHIBIT IT
20150
IF(IDEST.EQ.1)
GO TO 21000
G8 PG.64
IF(OALT.GT.ENR OTA(ITYPE))
TOP OF NEXT PAGE
PAGE 58
GO TO 20176
BOTTOM OF PRECEDING PAGE
C**** USE TERMINAL DESCENT RATE
CALL
CALL UPKUNI(ICLASS,IEIGHT,IPERFR3,.01,RATE)
GO TO 20180
C**** USE ENROUTE DESCENT RATE
20176
CALL
CALL UPKUNI(ICLASS,IEIGHT,IPERFR2,.01,RATE)
C**** COMPARE DESCENT RATE IN FT/NM, THEN COMPUTE ACTUAL ALTITUDE
20180
RDESC=RATE/OSPD/60.
AALT=AMAX1(OALT-RDESC*TOTD,DALT)
IAALT=(AALT+500.)/1000.
AALT=IAALT *1000
IF(DALT.EQ.AAL T)
GO TO 20190
F15 PG.59
CALL
CALL HALT
IWHERE=20180
ENCODE(70,5435,TYBUF) IWHERE,IDENT,IT,DALT,AALT
IBY=0
CALL
CALL WRT218(TYBUF,IBY,IX1)
CALL
CALL READY
F15 PG.58
AALT=DALT
20190
CONTINUE
GO TO 21000
G8 PG.64
F16 PG.57
C**** A/C DESIRES TO CLIMB.IF A/C IS AN ARRIVAL PROHIBIT IT
20500
IF(IDEST.EQ.2)
GO TO 21000
G8 PG.64
C**** COMPUTE RATE OF CLIMB IN FT/NM 10000 FT
CALL
CALL UPKUNI(ICLASS,IEIGHT,IPERFR4,.01,RATE)
CALL
CALL UPKUNI(ICLASS,ISIX,IPERFS2,1.,SPEED)
RCFNM(1)=RATE/SPEED*60.
C**** COMPUTE RATE OF CLIMB IN FT/NM ABOVE 10000 FT
CALL
CALL UPKUNI(ICLASS,IEIGHT,IPERFR5,.01,RATE)
PAGE 60
CALL
CALL UPKUNI(ICLASS,ISIX,IPERFS3,1.,SPEED)
RCFNM(2)=RATE/SPEED*60.
C**** COMPUTE DISTANCE TO TRAVEL LATERALLY WHILE CLIMBING TO 10000 FT
20510
DST1=AMAX1(AMIN1(DALT,10000.)-OALT,0.)/RCFNM(1)
C**** COMPUTE DISTANCE TO TRAVEL LATERALLY WHILE CLIMBING ABOVE 10000 FT
DST2=AMAX1(DALT-AMAX1(OALT,10000.),0.)/RCFNM(2)
C**** ACTUAL ALTITUDE AT END OF PATH
AALT= AMIN1(AMAX1(TOTD-DST1,0.),DST2)*RCFNM(2)
AALT=AALT+AMIN1(TOTD,DST1)*RCFNM(1)+OALT
IAALT=(AALT+500.)/1000.
AALT=IAALT*1000
C**** IS THERE A CONFLICT WITH ARRIVAL SPACE
20600
IF(IASPCON.EQ. 0)
GO TO 21000
G8 PG.64
C**** AIRSPACE CHECK
20605
IF(AALT.GT.ARR MXA)
TOP OF NEXT PAGE

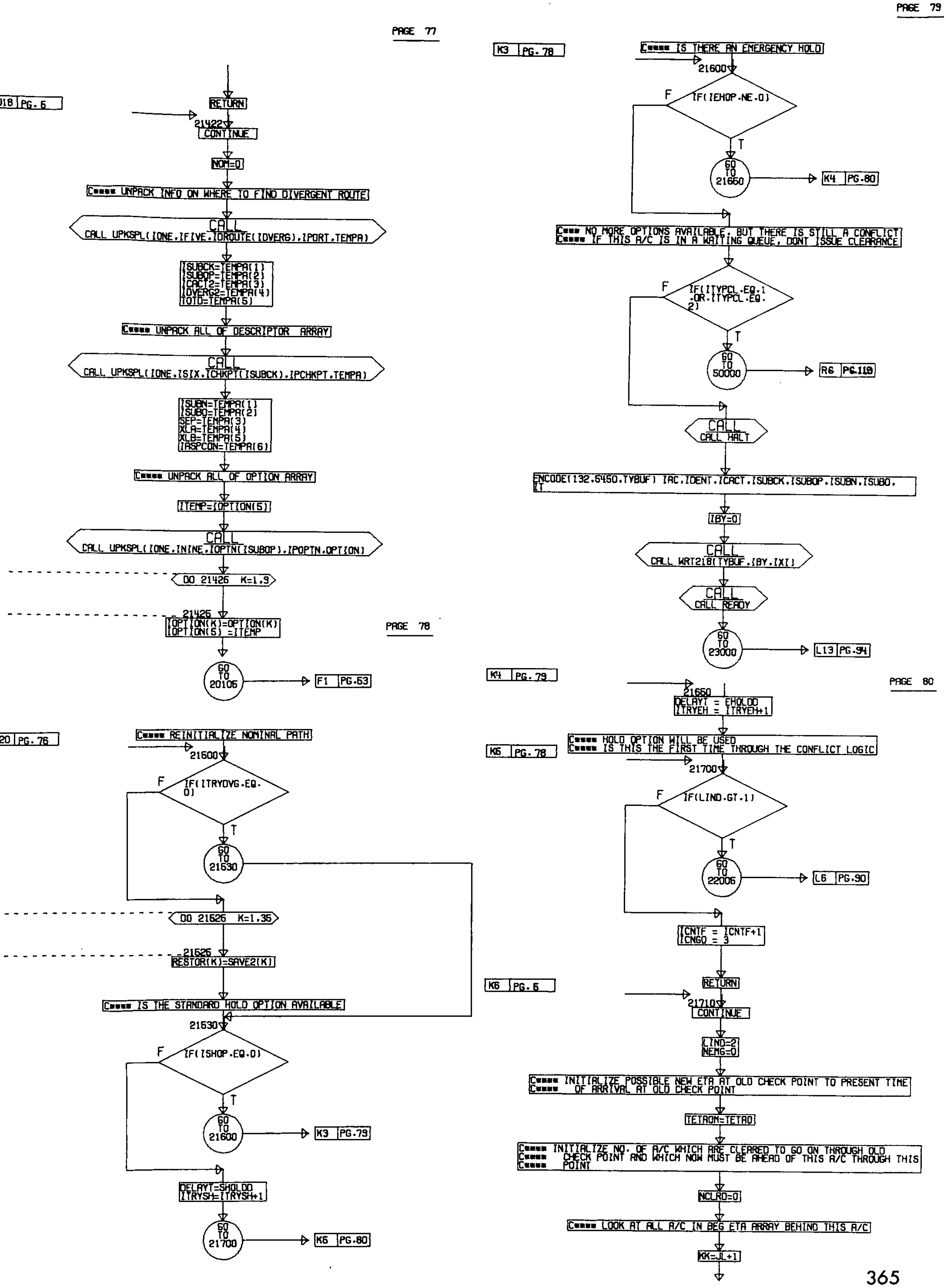

PAGE 61
GO TO 21000
G8 PG.64
BOTTOM OF PRECEDING PAGE
IF(AALT.LT.ARR MNA)
F
T
GO TO 21000
G8 PG.64
C**** IS THE ACTUAL ALTITUDE ONE OF THE ALTITUDE OPTIONS IN ALTOPT
DO 20610 KK=1.3
TEMP=ALTOPT(KK.IDALT.ITYPE)
IF(TEMP.EQ.0.)
GO TO 20615
G1 PG.62
IF(AALT.EQ.TEMP)
GO TO 20750
G7 PG.64
PAGE 62
20610
CONTINUE
C**** THE AIRCRAFT WANTS TO FLY THROUGH ARRIVAL AIRSPACE
C**** IS THERE AN ALTITUDE CONFLICT WITH ARRIVALS AT THIS ALTITUDE
1 PG.61
20615
CONTINUE
K=NASPACE(IASPCON)
IF(K.EQ.0)
GO TO 21000
G8 PG.64
DO 20625 KK=1.K
IF(AALT.EQ.ASPACE2(IASPCON.KK))
GO TO 20650
20625
CONTINUE
C**** NO CONFLICT WITH ARRIVALS A/C CAN USE THIS ALTITUDE
GO TO 21000
G8 PG.64
C**** THIS ALTITUDE IS BEING USED BY AN ARRIVAL. TRY A DIFF ALTITUDE
20650

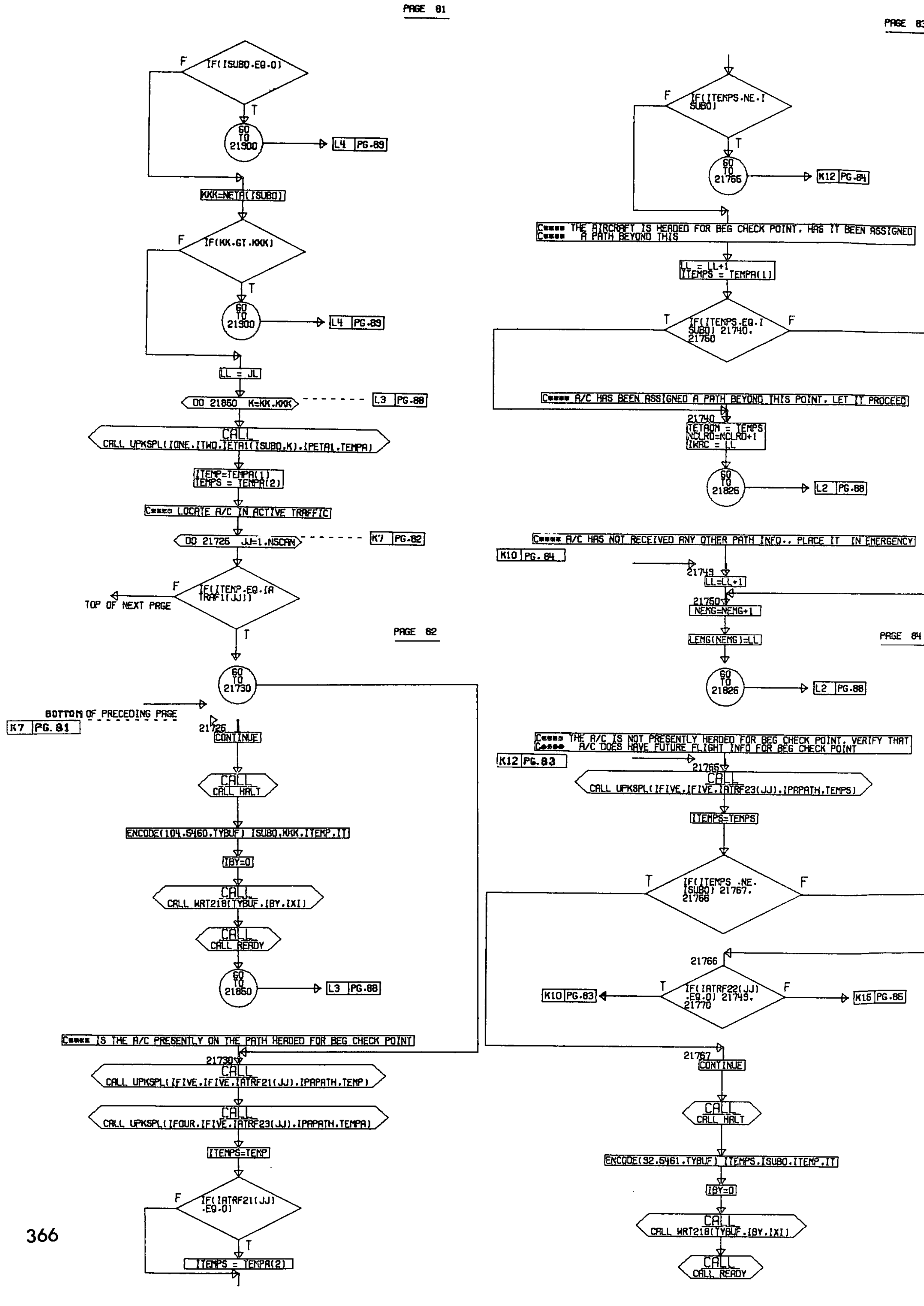

PAGE 63
CONTINUE
DO 20675 KK=1.3
TEMP=ALTOPT(KK.IDALT.ITYPE)
IF(TEMP.EQ.0.)
F
T
GO TO 20680
IF(TEMP.LT.AALT)
GO TO 20700
G6 PG.64
20675
CONTINUE
20680
CONTINUE
CALL
CALL HALT
ENCODE(144.6440.TYBUF) IAC.IDENT.ICACT.ISUBCK.ISUBOP.IDALT.IT
IBY=0
CALL
CALL WRT218(TYBUF.IBY.IXI)
PAGE 64
CALL
CALL READY
GO TO 20750
G6 PG.63
G7 PG.61
20700
AALT=ALTOPT(KK.IDALT.ITYPE)
20750
CONTINUE
IWCHAD=KK
C**** COMPUTE DELAY IN APPLYING SPEED CHANGE
G8 PG.57,57,59,59,60,61
21000
IF(OSPD.GT.0.)
TX=DST3/OSPD
C**** CALL SUBROUTINE TO COMPUTE TETA(TIME TO ARRIVE AT CHECK POINT)
G9 PG.69,73,73
21010
ITRY=1
CALL
CALL ETACOMP
C**** SHOULD A CONFLICT CHECK BE MADE
IF(ITYPCL.NE.5)
GO TO 21011
G10 PG.66

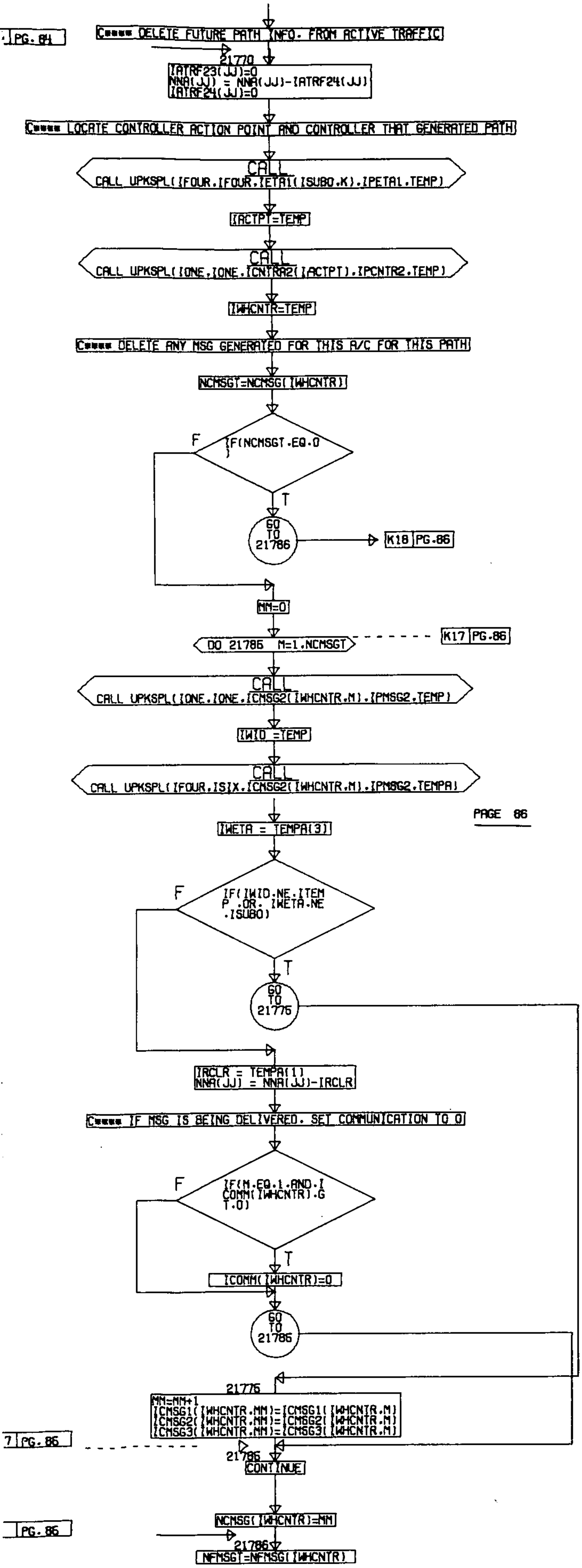

PAGE 65
CONFLCT = 0.
IWHICH = NETA(ISUBN)
GO TO 23000
L13 PG.94
G10 PG.64
21011
CALL
CALL CONCHK
NOM=NOM+1
IF(NOM.GT.1)
GO TO 21016
G13 PG.66
Cxxxx SAVE THE NOMINAL PATH INFO. FOR POSSIBLE FUTURE REFERENCE
KALTOP =IKCHAO
DO 21012 K=1,35
21012
SAVE(K) = RESTOR(K)
IF(ITRYOVG.GT.0)
TOP OF NEXT PAGE
GO TO 21016
PAGE 66
BOTTOM OF PRECEDING PAGE
DO 21013 K=1,35
21013
SAVE2(K) = RESTOR(K)
G13 PG.65 ,68
Cxxxx IS THERE A CONFLICT
21015
IF(CONFLCT.NE.0.)
GO TO 21020
G14 PG.67
Cxxxx NO CONFLICT
Cxxxx HAS A 360 PREVIOUSLY TRIED
IF(LIND.EQ.0)
GO TO 23000
L13 PG.94
Cxxxx A 360 IS NEEDED TOO RESOLVE THE CONFLICT ON THE FIRST PASS
Cxxxx THROUGH THE CONFLICT LOGIC. BUT NO 360 IS NEEDED FOR THE SECOND
HT2=0
PAGE 67
GO TO 22026
L7 PG.90
G14 PG.66
Cxxxx CONFLICT
Cxxxx IS A SPEED OPTION AVAILABLE
21020
IF(ISPOOP.EQ.0 .OR.IWHSPD.GT.2)
GO TO 21200
J1 PG.70
Cxxxx SPEED OPTION AVAILABLE
Cxxxx HAS A VARIATION ON SPEED BEEN TRIED
IF(IVARSPD.GT.1)
GO TO 21100
G16 PG.68
Cxxxx COMPUTE EARLIEST ETA AND LATEST ETA IF TX NE 0
IF(TX.EQ.0.)
GO TO 21026
G15 PG.68
PAGE 68
DST3=0.
TX=0.
ITRY=1
CALL
CALL ETACOMP
G15 PG.67
Cxxxx IS A VARIATION ON TIME APPLICATION OF SPEED PRACTICAL
21026
IF(TETAL.EQ.TETRE)
GO TO 21100
Cxxxx CALL SPEED CONTROL ROUTINE
ITRYSPD=1
IVARSPD=2
INDSPD=1
CALL
CALL SPDCNTR
GO TO 21015
G13 PG.66
G16 PG.67
Cxxxx IS THERE ANOTHER SPEED TO TRY
21100
IWHSPD=IWHSPD+1
IVARSPD = 1
IF(IWHSPD-2) 21116,21150,21200
G17 PG.69
J1 PG.70
G18 PG.69


PAGE 69
17 PG. 68
21115
TEMP=SPDSTD(IDSPD,ITYPE)
IF(TEMP.EQ.0.)
F
T
GO TO 21170
C***** TRY FASTER SPEED
DSPD= SPDSAV+SPDSTD(IDSPD,ITYPE)/3600.
GO TO 21160
C***** TRY SLOWER SPEED
18 PG. 68
21150
DSPD=SPDSAV-SPDSTD(IDSPD,ITYPE)/3600.
21160
TX=SAVE(8)
TX3=SAVE(34)
INDSPD=SAVE(35)
DST3=SAVE(13)
ITRYSPD=1
GO TO 21010
G9 PG.64
21170
IWHSPD = 3
C***** BECAUSE SPEED OPTION MIGHT HAVE BEEN TRIED, RESTORE NOMINAL PATH
PAGE 70
J1 PG. 67 ,68
21200
CONTINUE
IF(ITRYSPD.EQ.0)
GO TO 21210
DO 21205 K=1,35
21205
RESTOR(K) = SAVE(K)
ITRYSPD=0
C***** IS ALTITUDE OPTION AVAILABLE
21210
IF(IALTOP.EQ.0)
GO TO 21300
J10 PG.73
IF(KALTOP.GT.3)
GO TO 21300
J10 PG.73
PAGE 71
DO 21226 K=KALTOP,3
J6 PG.72
TEMP = ALTOPT(K,IALTOP,ITYPE)
IF(TEMP.EQ.0.)
GO TO 21226
J6 PG.72
IF(IDEST.EQ.1)
GO TO 21216
J4 PG.72
C***** ARRIVAL ALTITUDES
IF(AALT.LT.TEMP.AND.TEMP.LE.DALT)
GO TO 21250
J8 PG.72
GO TO 21225
J5 PG.72
PAGE 72
C***** DEPARTURE ALTITUDES
J4 PG. 71
21216
IF(AALT.GT.TEMP.AND.TEMP.GE.DALT)
GO TO 21250
J5 PG. 71 ,71
21225
CONTINUE
J6 PG. 71
21226
CONTINUE
C***** NO OTHER ALTITUDE OPTION IS AVAILABLE, RESET NOMINAL PATH
DO 21230 K=1,35
21230
RESTOR(K) = SAVE(K)
GO TO 21300
J10 PG.73
C***** TRY NEW ALTITUDE
J8 PG. 71
21250
AALT=TEMP
KALTOP = K
IF(IDEST.EQ.2)
TOP OF NEXT PAGE

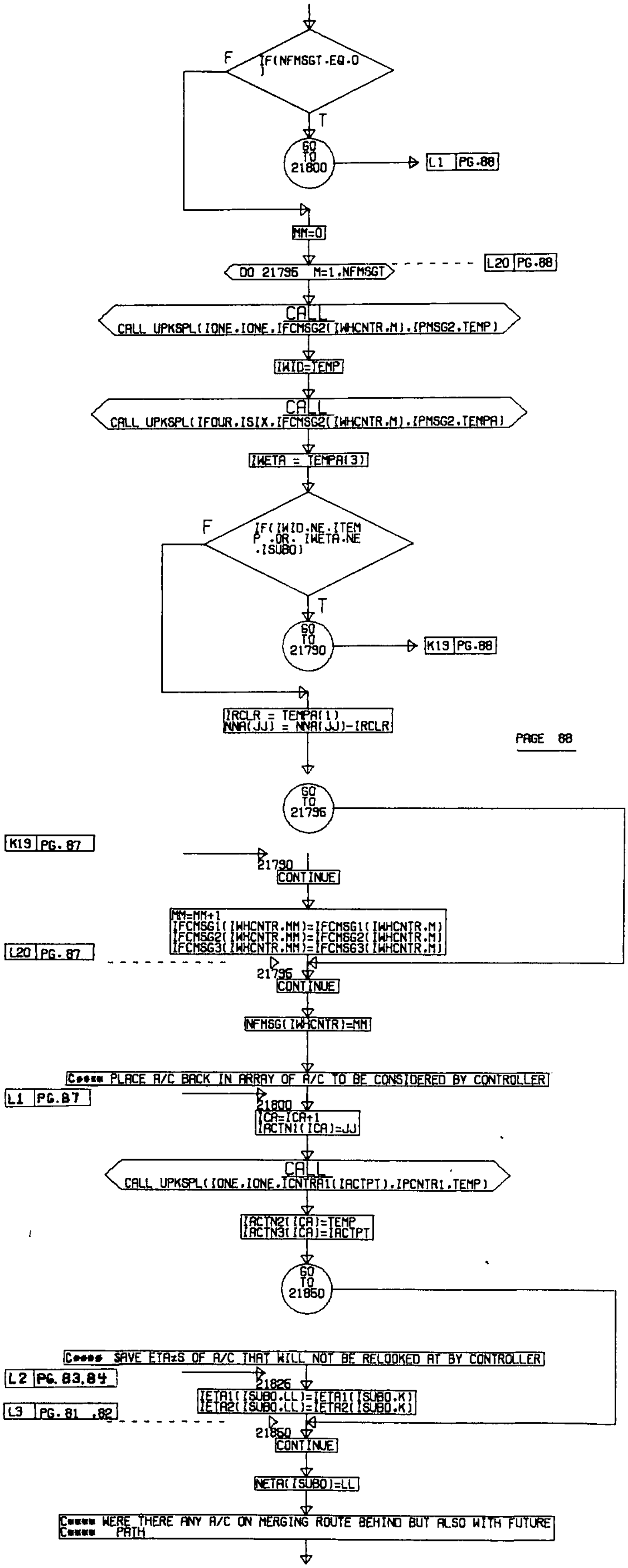

PAGE 73
GO TO 21260
BOTTOM OF PRECEDING PAGE
DST1=AMAX1(AMIN1(TEMP,10000.)-QALT,0.)/RCFNM(1)
DST2=AMAX1(TEMP-AMAX1(QALT,10000.),0.)/RCFNM(2)
C**** RECOMPUTE LATERAL DISTANCE TRAVELED WHILE CLIMBING
GO TO 21010
G9 PG.64
21260
DST1=0.
DST2=0.
GO TO 21010
G9 PG.64
J10 PG.70 ,70,72
C**** IS THERE AN ALTERNATE VECTOR TO TRY
21300
IF(IVECT2.EQ.0)
F
T
GO TO 21350
J13 PG.75
ITRYVEC=1
IVECT=IVECT2
C**** UNPACK INFO ON VECTOR PATH
CALL
CALL UPKSPL(IONE,ISEVEN,NNPATH(IVECT),IPNPATH,TEMPA)
PAGE 74
NSEGN=TEMPA(1)
IVECT2 = TEMPA(2)
NUMPATH=TEMPA(3)
DST3=0.
TOTD=0.
C**** COMPUTE TOTAL DISTANCE AND HOW MUCH DISTANCE SHOULD BE FLOWN
C**** BEFORE CHANGING SPEED
ISUBSPD=0
DO 21325 K=1,NSEGN
J12 PG.75
KP3=K+3
IWHCH=TEMPA(KP3)
CALL
CALL UPKWN1(IWHCH,ISIX,ISDST,10.,TEMPS)
TOTD=TOTD+TEMPS
CALL
CALL UPKSPL(IFIVE,IFIVE,ISKED(IWHCH),IPSKED,TEMP)
ITEMP = TEMP
IF(ITEMP.NE.ID SPD)
F
T
GO TO 21320
J11 PG.75
ISUBSPD=TEMP
GO TO 21325
J12 PG.75
J11 PG.74
PAGE 75
21320
IF(ISUBSPD.NE.0)
F
T
GO TO 21325
J12 PG.74 ,74
DST3=DST3+TEMPS
21325
CONTINUE
C**** TEST TO SEE IF SCHEDULE DICTATES WHERE SPD CHANGE SHOULD OCCUR
IF(ISUBSPD.EQ.0)
F
T
DST3=0.
TX=0.
GO TO 21000
G8 PG.64
J13 PG.75
C**** VECTOR PATH MAY HAVE BEEN TRIED, RESTORE NOMINAL PATH INFO
21350
IF(ITRYVEC.EQ.0)
F
T
GO TO 21400
J15 PG.76
PAGE 76
DO 21360 K=1,35
21360
RESTOR(K)=SAVE(K)
J16 PG.75
C**** IS THERE A DIVERGENT ROUTE TO BE TRIED
21400
IF(IDVERG2.EQ.0)
F
T
GO TO 21600
K20 PG.78
IF(ITRYDVG.GE.1)
F
T
GO TO 21420
C**** SAVE NOMINAL PATH BEFORE DIVERGENT PATH TRIED
DO 21416 K=1,35
21416
SAVE2(K)=SAVE(K)
21420
ITRYDVG = ITRYDVG+1
IDVERG = IDVERG2
ICNTF = ICNTF+1
ICNGO = 2


PAGE 77
J18 PG. 6
RETURN
21422
CONTINUE
NOM=0
C***** UNPACK INFO ON WHERE TO FIND DIVERGENT ROUTE
CALL
CALL UPKSPL(IONE,IFIVE,IOROUTE(IDVERG),IPORT,TEMPA)
ISUBCK=TEMPA(1)
ISUBOP=TEMPA(2)
ICACT2=TEMPA(3)
IDVERG2=TEMPA(4)
IOTD=TEMPA(5)
C***** UNPACK ALL OF DESCRIPTOR ARRAY
CALL
CALL UPKSPL(IONE,ISIX,ICHKPT(ISUBCK),IPCHKPT,TEMPA)
ISUBN=TEMPA(1)
ISUBO=TEMPA(2)
SEP=TEMPA(3)
XLA=TEMPA(4)
XLB=TEMPA(5)
IASPCON=TEMPA(6)
C***** UNPACK ALL OF OPTION ARRAY
ITEMP=IOPTION(5)
CALL
CALL UPKSPL(IONE,ININE,IOPTN(ISUBOP),IPOPTN,OPTION)
DO 21425 K=1,9
21425
IOPTION(K)=OPTION(K)
IOPTION(5) =ITEMP
GO TO 20105
F1 PG.53
PAGE 78
20 PG. 76
C***** REINITIALIZE NOMINAL PATH
21600
F
IF(ITRYDVG.EQ.0)
T
GO TO 21630
DO 21525 K=1,35
21525
RESTOR(K)=SAVE2(K)
C***** IS THE STANDARD HOLD OPTION AVAILABLE
21530
F
IF(ISHOP.EQ.0)
T
GO TO 21600
K3 PG.79
DELAYT=SHOLDD
ITRYSH=ITRYSH+1
GO TO 21700
K5 PG.80
PAGE 79
K3 PG. 78
C***** IS THERE AN EMERGENCY HOLD
21600
F
IF(IEHOP.NE.0)
T
GO TO 21650
K4 PG.80
C**** NO MORE OPTIONS AVAILABLE, BUT THERE IS STILL A CONFLICT
C***** IF THIS A/C IS IN A WAITING QUEUE, DONT ISSUE CLEARANCE
F
IF(ITYPCL.EQ.1 .OR.ITYPCL.EQ. 2)
T
GO TO 50000
R6 PG.119
CALL
CALL HALT
ENCODE(132,5450,TYBUF) IAC,IDENT,ICACT,ISUBCK,ISUBOP,ISUBN,ISUBO, IT
IBY=0
CALL
CALL WRT218(TYBUF,IBY,IXI)
CALL
CALL READY
GO TO 23000
L13 PG.94
PAGE 80
K4 PG. 79
21650
DELAYT = EHOLDD
ITRYEH = ITRYEH+1
K5 PG. 78
C***** HOLD OPTION WILL BE USED
C***** IS THIS THE FIRST TIME THROUGH THE CONFLICT LOGIC
21700
F
IF(LIND.GT.1)
T
GO TO 22006
L6 PG.90
ICNTF = ICNTF+1
ICNGO = 3
RETURN
K6 PG. 6
21710
CONTINUE
LIND=2
NEMG=0
C***** INITIALIZE POSSIBLE NEW ETA AT OLD CHECK POINT TO PRESENT TIME
C***** OF ARRIVAL AT OLD CHECK POINT
TETAOM=TETAO
C***** INITIALIZE NO. OF A/C WHICH ARE CLEARED TO GO ON THROUGH OLD
C***** CHECK POINT AND WHICH NOW MUST BE AHEAD OF THIS A/C THROUGH THIS
C***** POINT
NCLRD=0
C***** LOOK AT ALL A/C IN BEG ETA ARRAY BEHIND THIS A/C
KK=JL+1

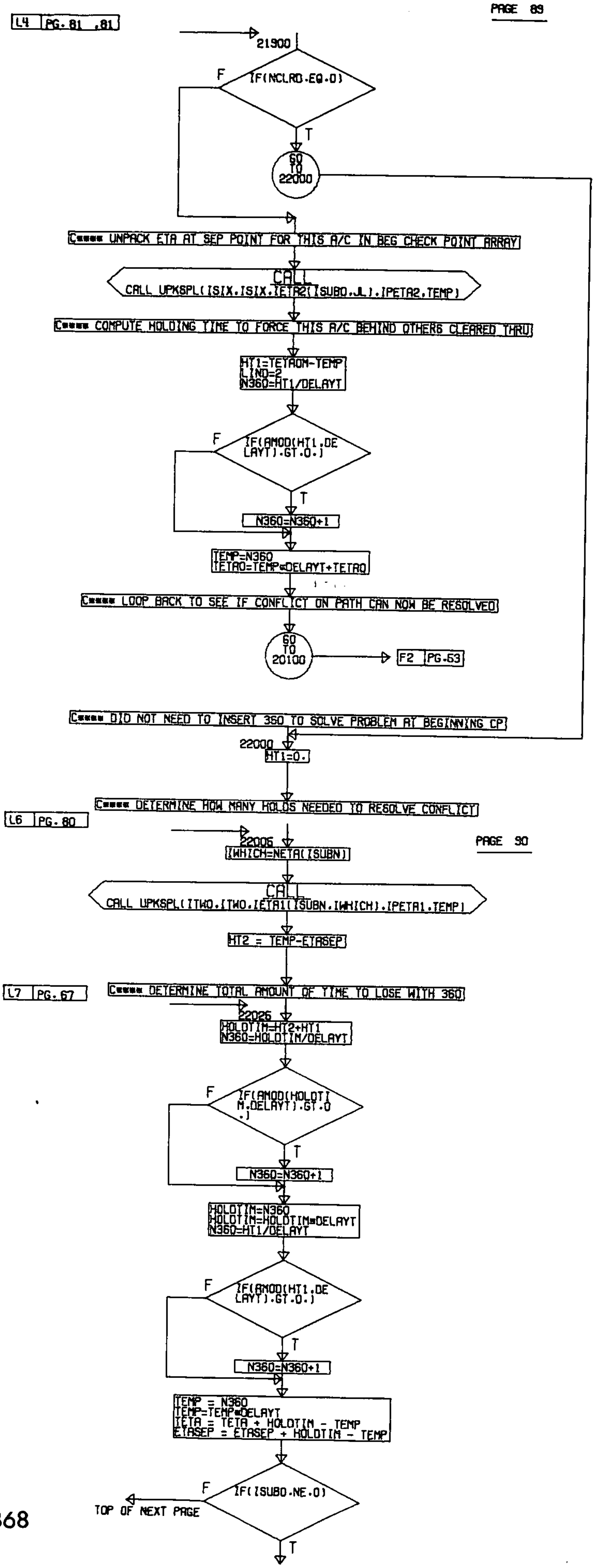

PAGE 81
IF(ISUBO.EQ.0)
F
T
GO TO 21900
L4 PG.89
KKK=NETA(ISUBO)
IF(KK.GT.KKK)
F
T
GO TO 21900
L4 PG.89
LL = JL
DO 21850 K=KK.KKK
L3 PG.88
CALL
CALL UPKSPL(IONE.ITWO.IETA1(ISUBO.K).IPETA1.TEMPA)
ITEMP=TEMPA(1)
TEMPS = TEMPA(2)
C***** LOCATE A/C IN ACTIVE TRAFFIC
DO 21725 JJ=1.NSCAN
K7 PG.82
IF(ITEMP.EQ.IATRAF1(JJ))
F
TOP OF NEXT PAGE
T
GO TO 21730
PAGE 82
BOTTOM OF PRECEDING PAGE
K7 PG.81
21726
CONTINUE
CALL
CALL HALT
ENCODE(104.5460.TYBUF) ISUBO.KKK.ITEMP.IT
IBY=0
CALL
CALL WRT218(TYBUF.IBY.IXI)
CALL
CALL READY
GO TO 21850
L3 PG.88
C***** IS THE A/C PRESENTLY ON THE PATH HEADED FOR BEG CHECK POINT
21730
CALL
CALL UPKSPL(IFIVE.IFIVE.IATRF21(JJ).IPAPATH.TEMP)
CALL
CALL UPKSPL(IFOUR.IFIVE.IATRF23(JJ).IPAPATH.TEMPA)
ITEMPS=TEMP
IF(IATRF21(JJ).EQ.0)
F
T
ITEMPS = TEMPA(2)
PAGE 83
IF(ITEMPS.NE.ISUBO)
F
T
GO TO 21766
K12 PG.84
C***** THE AIRCRAFT IS HEADED FOR BEG CHECK POINT. HAS IT BEEN ASSIGNED
C***** A PATH BEYOND THIS
LL = LL+1
ITEMPS = TEMPA(1)
IF(ITEMPS.EQ.ISUBO) 21740.21750
T
F
C***** A/C HAS BEEN ASSIGNED A PATH BEYOND THIS POINT. LET IT PROCEED
21740
IETAOM = TEMPS
NCLRD=NCLRD+1
IKAC = LL
GO TO 21826
L2 PG.88
C***** A/C HAS NOT RECEIVED ANY OTHER PATH INFO.. PLACE IT IN EMERGENCY
K10 PG. 84
21749
LL=LL+1
21750
NEMG=NEMG+1
LEMG(NEMG)=LL
PAGE 84
GO TO 21826
L2 PG.88
C***** THE A/C IS NOT PRESENTLY HEADED FOR BEG CHECK POINT. VERIFY THAT
C***** A/C DOES HAVE FUTURE FLIGHT INFO FOR BEG CHECK POINT
K12 PG.83
21765
CALL
CALL UPKSPL(IFIVE.IFIVE.IATRF23(JJ).IPAPATH.TEMPS)
ITEMPS=TEMPS
IF(ITEMPS .NE. ISUBO) 21767.21766
T
F
21766
IF(IATRF22(JJ).EQ.0) 21749.21770
T
F
K10 PG.83
K15 PG.85
21767
CONTINUE
CALL
CALL HALT
ENCODE(92.5461.TYBUF) ITEMPS.ISUBO.ITEMP.IT
IBY=0
CALL
CALL WRT218(TYBUF.IBY.IXI)
CALL
CALL READY



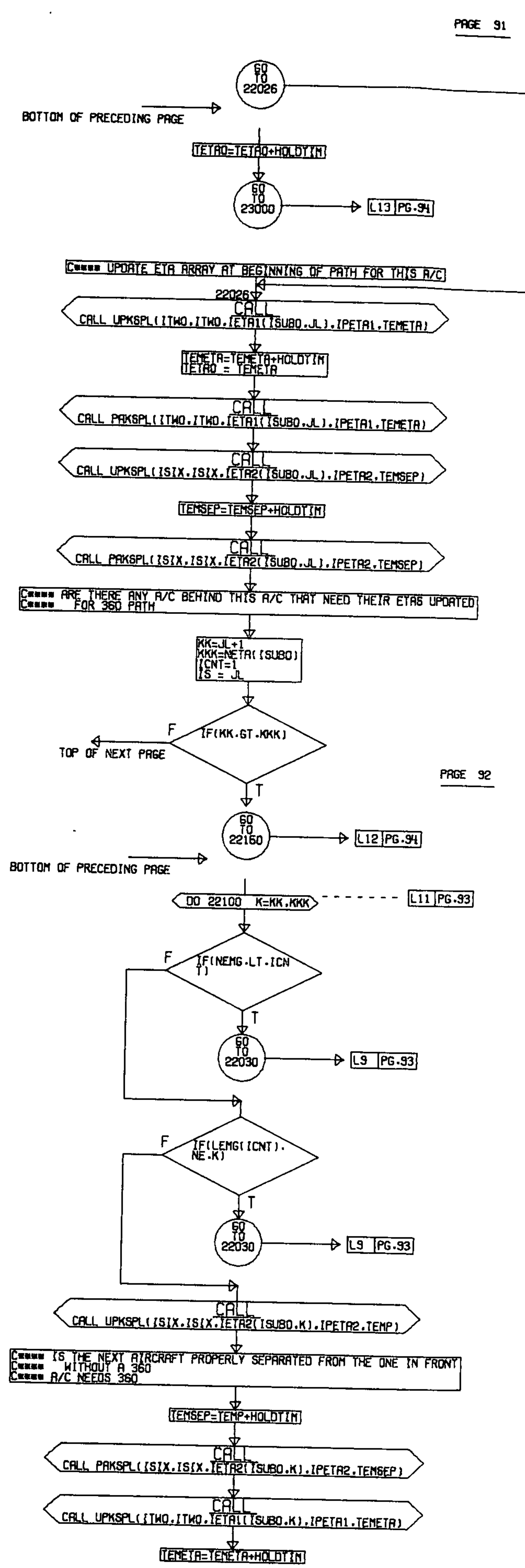
PG. 84
C**** DELETE FUTURE PATH INFO. FROM ACTIVE TRAFFIC
21770
IATRF23(JJ)=0
NNA(JJ) = NNA(JJ)-IATRF24(JJ)
IATRF24(JJ)=0
C**** LOCATE CONTROLLER ACTION POINT AND CONTROLLER THAT GENERATED PATH
CALL
CALL UPKSPL(IFOUR,IFOUR,IETA1(ISUBO,K),IPETA1,TEMP)
IACTPT=TEMP
CALL
CALL UPKSPL(IONE,IONE,ICNTRA2(IACTPT),IPCNTR2,TEMP)
IWHCNTR=TEMP
C**** DELETE ANY MSG GENERATED FOR THIS A/C FOR THIS PATH
NCMSGT=NCMSG(IWHCNTR)
F
IF(NCMSGT.EQ.0)
T
GO TO 21786
K18 PG.86
MM=0
DO 21785 M=1,NCMSGT
K17 PG.86
CALL
CALL UPKSPL(IONE,IONE,ICMSG2(IWHCNTR,M),IPMSG2,TEMP)
IWID =TEMP
CALL
CALL UPKSPL(IFOUR,ISIX,ICMSG2(IWHCNTR,M),IPMSG2,TEMPA)
PAGE 86
IWETA = TEMPA(3)
F
IF(IWID.NE.ITEMP .OR. IWETA.NE.ISUBO)
T
GO TO 21775
IRCLR = TEMPA(1)
NNA(JJ) = NNA(JJ)-IRCLR
C**** IF MSG IS BEING DELIVERED, SET COMMUNICATION TO 0
F
IF(M.EQ.1.AND.ICOMM(IWHCNTR).GT.0)
T
ICOMM(IWHCNTR)=0
GO TO 21785
21775
MM=MM+1
ICMSG1(IWHCNTR,MM)=ICMSG1(IWHCNTR,M)
ICMSG2(IWHCNTR,MM)=ICMSG2(IWHCNTR,M)
ICMSG3(IWHCNTR,MM)=ICMSG3(IWHCNTR,M)
7 PG.85
21785
CONTINUE
NCMSG(IWHCNTR)=MM
PG.85
21786
NFMSGT=NFMSG(IWHCNTR)



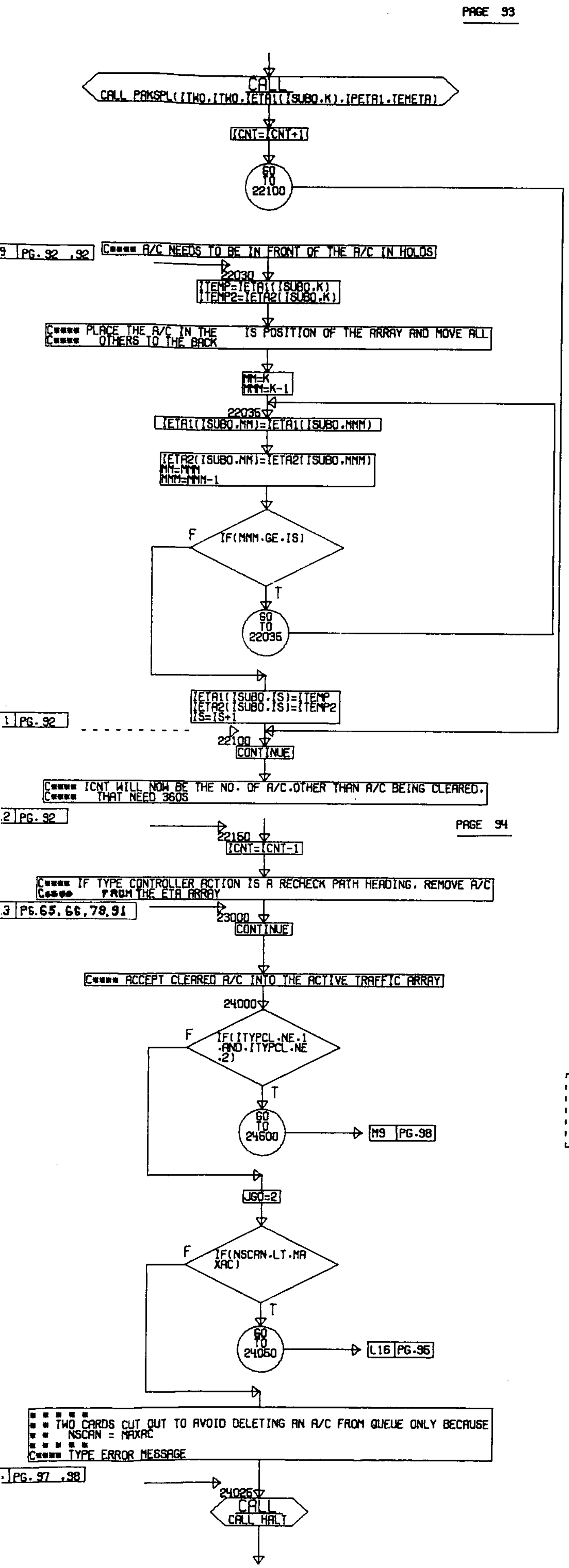
F
IF(NFMSGT.EQ.0)
T
GO TO 21800
L1 PG.88
MM=0
DO 21795 M=1,NFMSGT
L20 PG.88
CALL
CALL UPKSPL(IONE,IONE,IFCMSG2(IWHCNTR,M),IPMSG2,TEMP)
IWID=TEMP
CALL
CALL UPKSPL(IFOUR,ISIX,IFCMSG2(IWHCNTR,M),IPMSG2,TEMPA)
IWETA = TEMPA(3)
F
IF(IWID.NE.ITEMP .OR. IWETA.NE.ISUBO)
T
GO TO 21790
K19 PG.88
IRCLR = TEMPA(1)
NNA(JJ) = NNA(JJ)-IRCLR
PAGE 88
GO TO 21795
K19 PG.87
21790
CONTINUE
MM=MM+1
IFCMSG1(IWHCNTR,MM)=IFCMSG1(IWHCNTR,M)
IFCMSG2(IWHCNTR,MM)=IFCMSG2(IWHCNTR,M)
IFCMSG3(IWHCNTR,MM)=IFCMSG3(IWHCNTR,M)
L20 PG.87
21795
CONTINUE
NFMSG(IWHCNTR)=MM
C**** PLACE A/C BACK IN ARRAY OF A/C TO BE CONSIDERED BY CONTROLLER
L1 PG.87
21800
ICA=ICA+1
IACTN1(ICA)=JJ
CALL
CALL UPKSPL(IONE,IONE,ICNTRA1(IACTPT),IPCNTR1,TEMP)
IACTN2(ICA)=TEMP
IACTN3(ICA)=IACTPT
GO TO 21850
C**** SAVE ETA=S OF A/C THAT WILL NOT BE RELOOKED AT BY CONTROLLER
L2 PG. 83,84
21825
IETA1(ISUBO,LL)=IETA1(ISUBO,K)
IETA2(ISUBO,LL)=IETA2(ISUBO,K)
L3 PG. 81 ,82
21850
CONTINUE
NETA(ISUBO)=LL
C**** WERE THERE ANY A/C ON MERGING ROUTE BEHIND BUT ALSO WITH FUTURE
C**** PATH

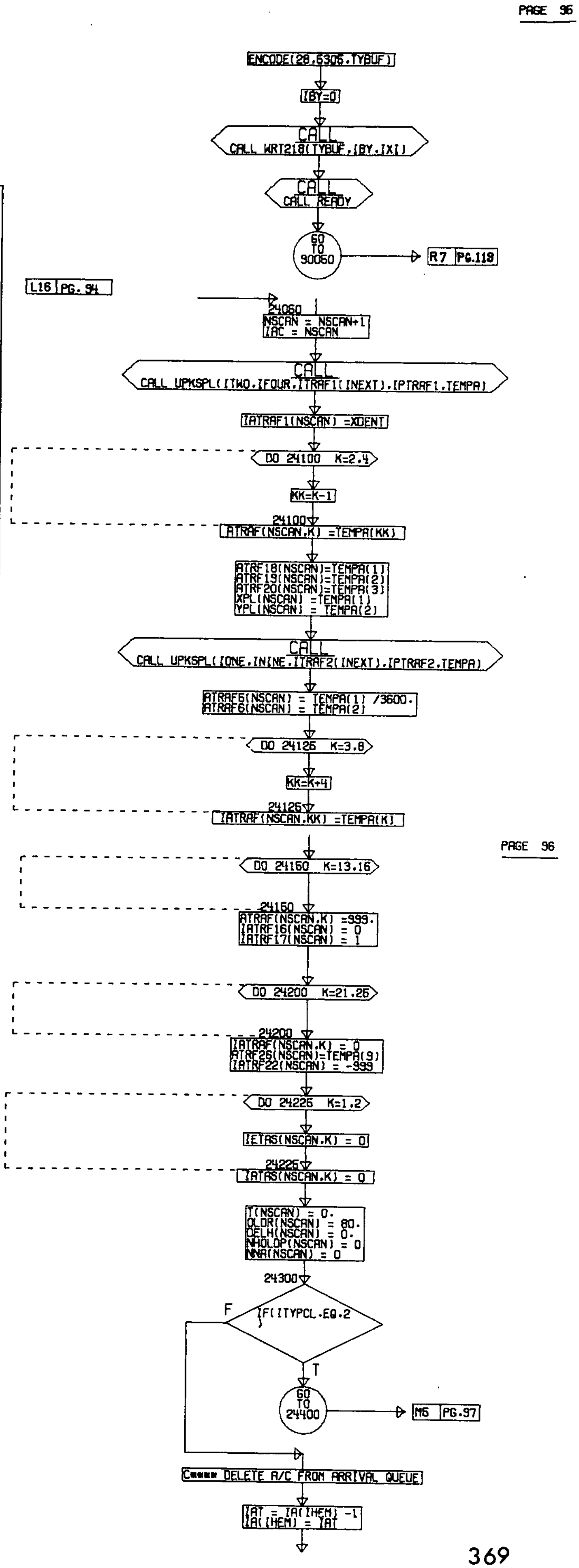


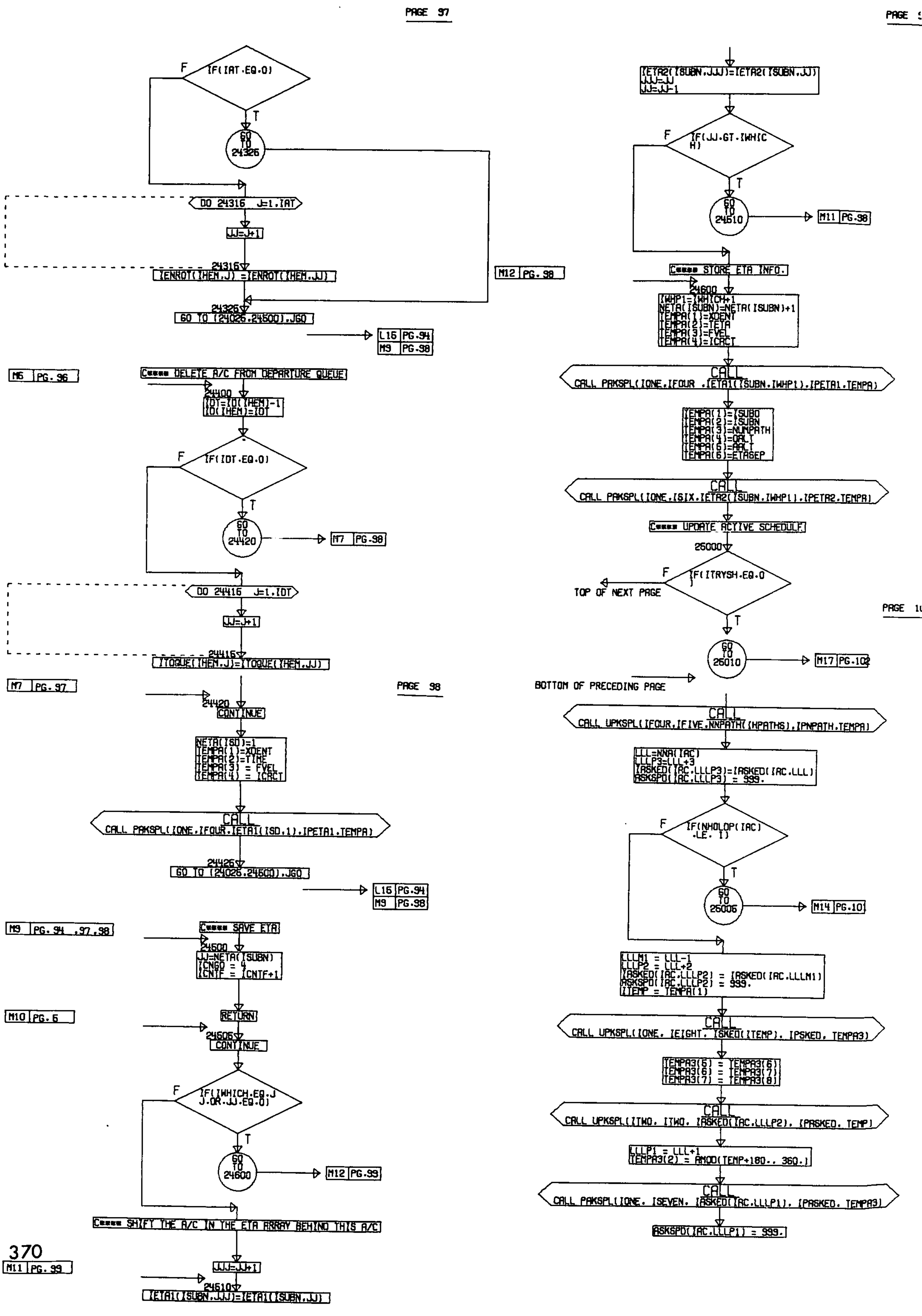
PAGE 89
L4 PG. 81 ,81
21900
IF(NCLRD.EQ.0)
F
T
GO TO 22000
C***** UNPACK ETA AT SEP POINT FOR THIS A/C IN BEG CHECK POINT ARRAY
CALL
CALL UPKSPL(ISIX,ISIX,IETA2(ISUBO,JL),IPETA2,TEMP)
C***** COMPUTE HOLDING TIME TO FORCE THIS A/C BEHIND OTHERS CLEARED THRU
HT1=TETAOM-TEMP
LIND=2
N360=HT1/DELAYT
IF(AMOD(HT1,DELAYT).GT.0.)
N360=N360+1
TEMP=N360
TETAO=TEMP*DELAYT+TETAO
C***** LOOP BACK TO SEE IF CONFLICT ON PATH CAN NOW BE RESOLVED
GO TO 20100
F2 PG.53
C***** DID NOT NEED TO INSERT 360 TO SOLVE PROBLEM AT BEGINNING CP
22000
HT1=0.
C***** DETERMINE HOW MANY HOLDS NEEDED TO RESOLVE CONFLICT
L6 PG. 80
22005
PAGE 90
IWHICH=NETA(ISUBN)
CALL
CALL UPKSPL(ITWO,ITWO,IETA1(ISUBN,IWHICH),IPETA1,TEMP)
HT2 = TEMP-ETASEP
L7 PG. 67
C***** DETERMINE TOTAL AMOUNT OF TIME TO LOSE WITH 360
22026
HOLDTIM=HT2+HT1
N360=HOLDTIM/DELAYT
IF(AMOD(HOLDTIM,DELAYT).GT.0.)
N360=N360+1
HOLDTIM=N360
HOLDTIM=HOLDTIM*DELAYT
N360=HT1/DELAYT
IF(AMOD(HT1,DELAYT).GT.0.)
N360=N360+1
TEMP = N360
TEMP=TEMP*DELAYT
IETA = TETA + HOLDTIM - TEMP
ETASEP = ETASEP + HOLDTIM - TEMP
IF(ISUBO.NE.0)
TOP OF NEXT PAGE

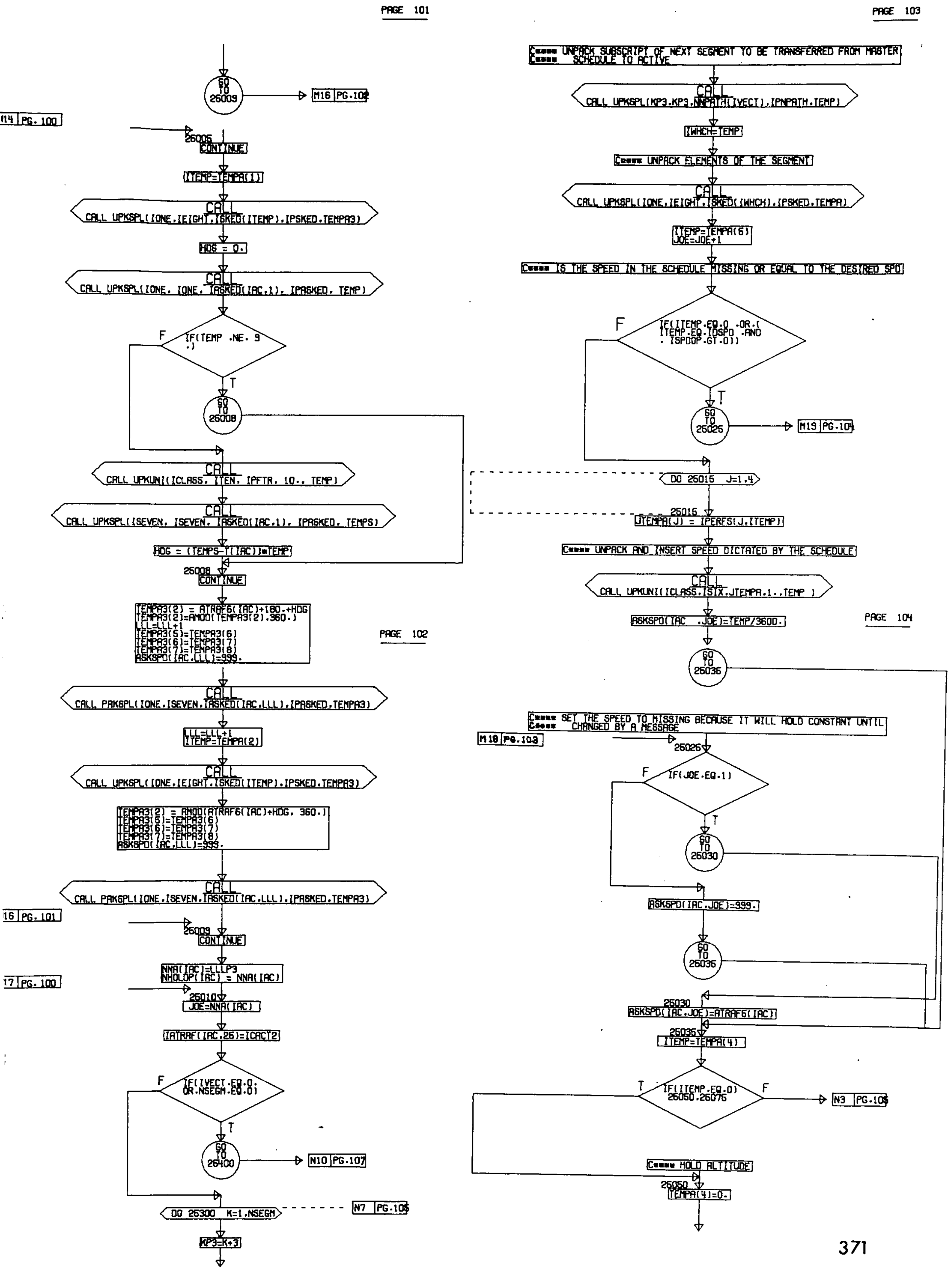


PAGE 91
GO TO 22026
BOTTOM OF PRECEDING PAGE
TETAO=TETAO+HOLDTIM
GO TO 23000
L13 PG.94
C***** UPDATE ETA ARRAY AT BEGINNING OF PATH FOR THIS A/C
22026
CALL
CALL UPKSPL(ITWO,ITWO,IETA1(ISUBO,JL),IPETA1,TEMETA)
TEMETA=TEMETA+HOLDTIM
TETAO = TEMETA
CALL
CALL PAKSPL(ITWO,ITWO,IETA1(ISUBO,JL),IPETA1,TEMETA)
CALL
CALL UPKSPL(ISIX,ISIX,IETA2(ISUBO,JL),IPETA2,TEMSEP)
TEMSEP=TEMSEP+HOLDTIM
CALL
CALL PAKSPL(ISIX,ISIX,IETA2(ISUBO,JL),IPETA2,TEMSEP)
C***** ARE THERE ANY A/C BEHIND THIS A/C THAT NEED THEIR ETAS UPDATED
C***** FOR 360 PATH
KK=JL+1
KKK=NETA(ISUBO)
ICNT=1
IS = JL
IF(KK.GT.KKK)
TOP OF NEXT PAGE
PAGE 92
GO TO 22150
L12 PG.94
BOTTOM OF PRECEDING PAGE
DO 22100 K=KK,KKK
L11 PG.93
IF(NEMG.LT.ICNT)
GO TO 22030
L9 PG.93
IF(LEMG(ICNT).NE.K)
GO TO 22030
L9 PG.93
CALL
CALL UPKSPL(ISIX,ISIX,IETA2(ISUBO,K),IPETA2,TEMP)
C***** IS THE NEXT AIRCRAFT PROPERLY SEPARATED FROM THE ONE IN FRONT
C***** WITHOUT A 360
C***** A/C NEEDS 360
TEMSEP=TEMP+HOLDTIM
CALL
CALL PAKSPL(ISIX,ISIX,IETA2(ISUBO,K),IPETA2,TEMSEP)
CALL
CALL UPKSPL(ITWO,ITWO,IETA1(ISUBO,K),IPETA1,TEMETA)
TEMETA=TEMETA+HOLDTIM

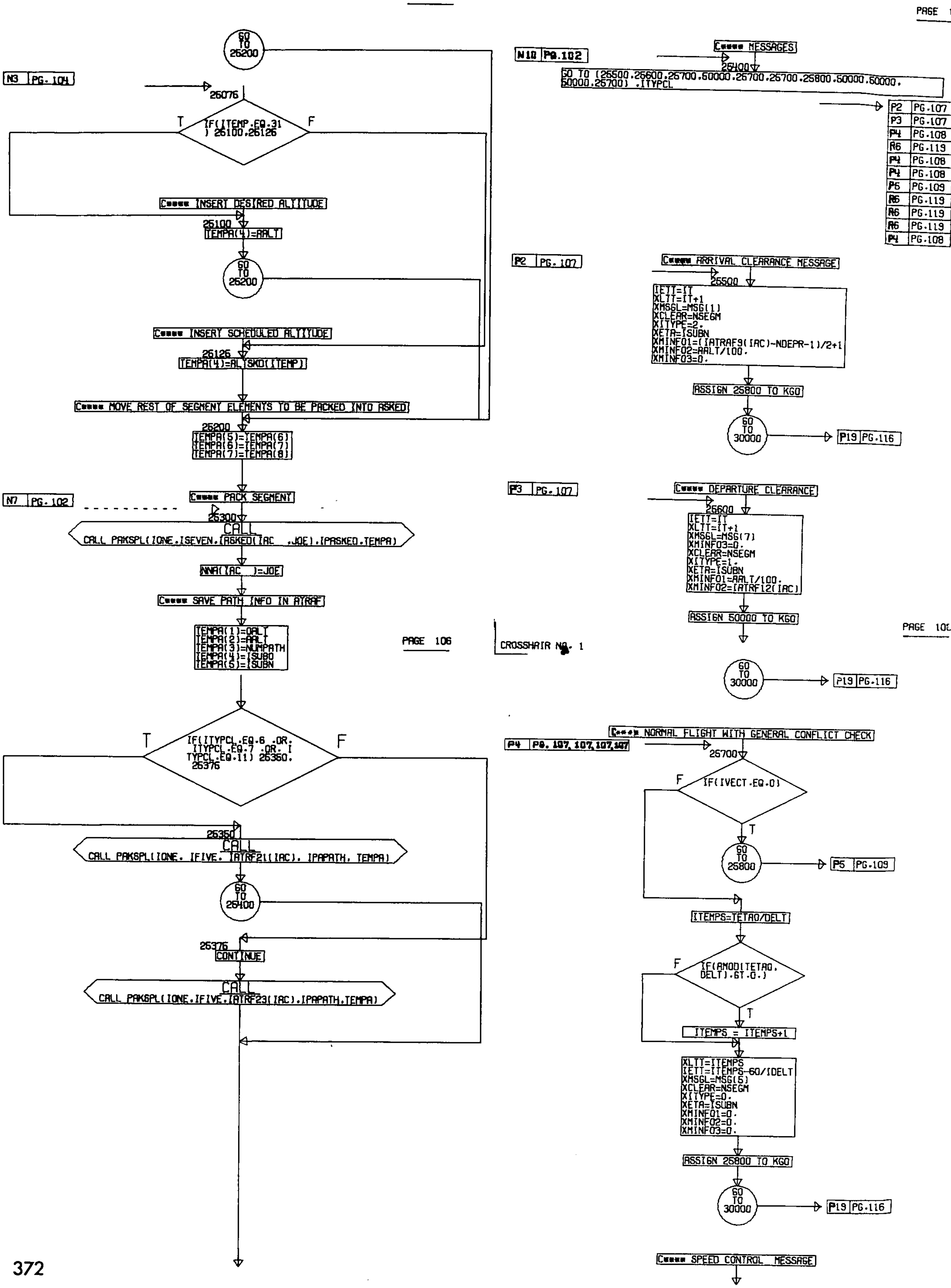

PAGE 93
CALL
CALL PAKSPL(ITWO,ITWO,IETA1(ISUBO,K),IPETA1,TEMETA)
ICNT=ICNT+1
GO TO 22100
PG. 92 ,92
C**** A/C NEEDS TO BE IN FRONT OF THE A/C IN HOLDS
22030
ITEMP=IETA1(ISUBO,K)
ITEMP2=IETA2(ISUBO,K)
C**** PLACE THE A/C IN THE IS POSITION OF THE ARRAY AND MOVE ALL
C**** OTHERS TO THE BACK
MM=K
MMM=K-1
22035
IETA1(ISUBO,MM)=IETA1(ISUBO,MMM)
IETA2(ISUBO,MM)=IETA2(ISUBO,MMM)
MM=MMM
MMM=MMM-1
IF(MMM.GE.IS)
F
T
GO TO 22035
IETA1(ISUBO,IS)=ITEMP
IETA2(ISUBO,IS)=ITEMP2
IS=IS+1
1 PG. 92
22100
CONTINUE
C**** ICNT WILL NOW BE THE NO. OF A/C OTHER THAN A/C BEING CLEARED,
C**** THAT NEED 3605
2 PG. 92
PAGE 94
22150
ICNT=ICNT-1
C**** IF TYPE CONTROLLER ACTION IS A RECHECK PATH HEADING, REMOVE A/C
C**** FROM THE ETA ARRAY
3 PG. 65, 66, 78, 91
23000
CONTINUE
C**** ACCEPT CLEARED A/C INTO THE ACTIVE TRAFFIC ARRAY
24000
IF(ITYPCL.NE.1 .AND.ITYPCL.NE .2)
F
T
GO TO 24600
M9 PG.98
JGO=2
IF(NSCAN.LT.MA XAC)
F
T
GO TO 24050
L16 PG.95
* * * * *
* * TWO CARDS CUT OUT TO AVOID DELETING AN A/C FROM QUEUE ONLY BECAUSE
* * NSCAN = MAXAC
* * * * *
C**** TYPE ERROR MESSAGE
PG. 97 ,98
24025
CALL
CALL HALT

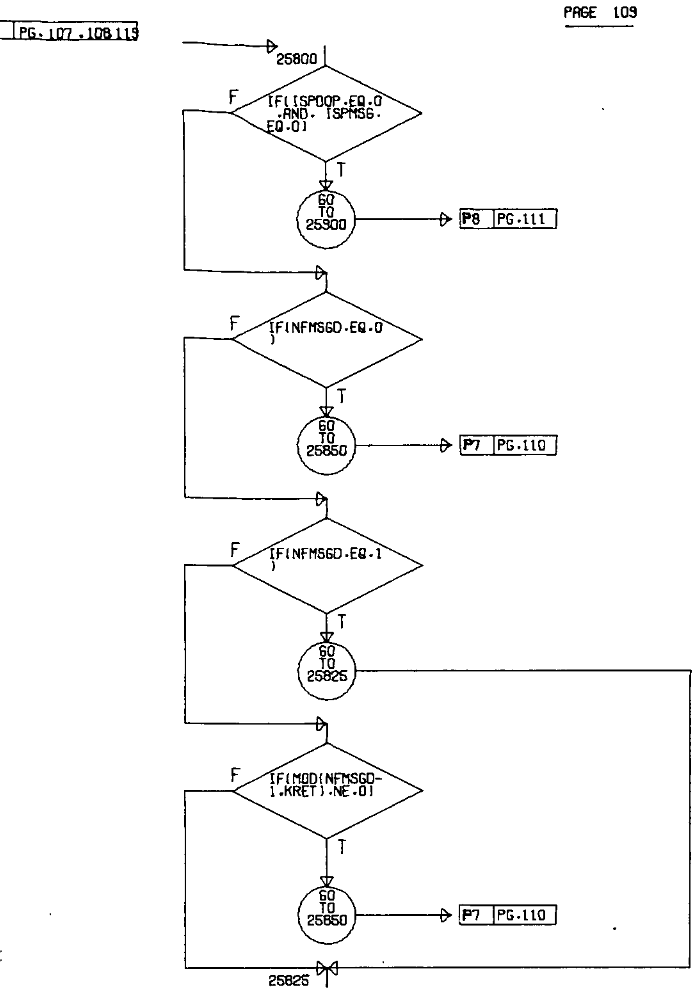

PAGE 95
ENCODE(28,6305,TYBUF)
IBY=0
CALL
CALL WRT218(TYBUF,IBY,IXI)
CALL
CALL READY
GO TO 30050
R7 PG.119
L16 PG. 94
24050
NSCAN = NSCAN+1
IAC = NSCAN
CALL
CALL UPKSPL(ITWO,IFOUR,ITRAF1(INEXT),IPTRAF1,TEMPA)
IATRAF1(NSCAN) =XDENT
DO 24100 K=2,4
KK=K-1
24100
ATRAF(NSCAN,K) =TEMPA(KK)
ATRF18(NSCAN)=TEMPA(1)
ATRF19(NSCAN)=TEMPA(2)
ATRF20(NSCAN)=TEMPA(3)
XPL(NSCAN) =TEMPA(1)
YPL(NSCAN) = TEMPA(2)
CALL
CALL UPKSPL(IONE,ININE,ITRAF2(INEXT),IPTRAF2,TEMPA)
ATRAF5(NSCAN) = TEMPA(1) /3600.
ATRAF6(NSCAN) = TEMPA(2)
DO 24125 K=3,8
KK=K+4
24125
IATRAF(NSCAN,KK) =TEMPA(K)
PAGE 96
DO 24150 K=13,16
24150
ATRAF(NSCAN,K) =999.
IATRF16(NSCAN) = 0
IATRF17(NSCAN) = 1
DO 24200 K=21,26
24200
IATRAF(NSCAN,K) = 0
ATRF26(NSCAN)=TEMPA(9)
IATRF22(NSCAN) = -999
DO 24225 K=1,2
IETAS(NSCAN,K) = 0
24225
IATAS(NSCAN,K) = 0
T(NSCAN) = 0.
OLDR(NSCAN) = 80.
DELH(NSCAN) = 0.
NHOLDP(NSCAN) = 0
NNA(NSCAN) = 0
24300
IF(ITYPCL.EQ.2)
F
T
GO TO 24400
M6 PG.97
C**** DELETE A/C FROM ARRIVAL QUEUE
IAT = IA(IHEM) -1
IA(IHEM) = IAT

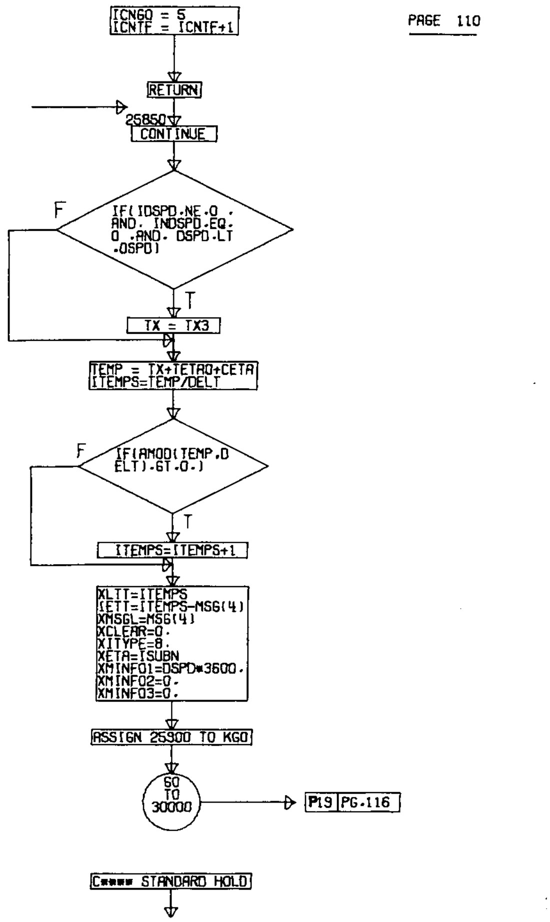

PAGE 97
IF(IAT.EQ.0)
F
T
GO TO 24326
DO 24316 J=1,IAT
JJ=J+1
24316
IENROT(IHEM,J) =IENROT(IHEM,JJ)
M12 PG. 98
24326
GO TO (24026,24500),JGO
L15 PG.94
M9 PG.98
M6 PG. 96
C**** DELETE A/C FROM DEPARTURE QUEUE
24400
IDT=ID(IHEM)-1
ID(IHEM)=IDT
IF(IDT.EQ.0)
GO TO 24420
M7 PG.98
DO 24416 J=1,IDT
JJ=J+1
24416
ITOQUE(IHEM,J)=ITOQUE(IHEM,JJ)
M7 PG. 97
PAGE 98
24420
CONTINUE
NETA(ISD)=1
TEMPA(1)=XDENT
TEMPA(2)=TIME
TEMPA(3) = FVEL
TEMPA(4) = ICRCT
CALL
CALL PAKSPL(IONE,IFOUR,IETA1(ISD,1),IPETA1,TEMPA)
24426
GO TO (24026,24500),JGO
L15 PG.94
M9 PG.98
M9 PG. 94 ,97,98
C**** SAVE ETA
24500
JJ=NETA(ISUBN)
ICNGO = 4
ICNTF = ICNTF+1
RETURN
M10 PG. 6
24505
CONTINUE
IF(IWHICH.EQ.JJ.OR.JJ.EQ.0)
GO TO 24600
M12 PG.99
C**** SHIFT THE A/C IN THE ETA ARRAY BEHIND THIS A/C
M11 PG. 99
JJJ=JJ+1
24510
IETA1(ISUBN,JJJ)=IETA1(ISUBN,JJ)
PAGE 99
IETA2(ISUBN,JJJ)=IETA2(ISUBN,JJ)
JJJ=JJ
JJ=JJ-1
IF(JJ.GT.IWHICH)
GO TO 24510
M11 PG.98
C**** STORE ETA INFO.
24600
IWHP1=IWHICH+1
NETA(ISUBN)=NETA(ISUBN)+1
TEMPA(1)=XDENT
TEMPA(2)=TETA
TEMPA(3)=FVEL
TEMPA(4)=ICRCT
CALL
CALL PAKSPL(IONE,IFOUR ,IETA1(ISUBN,IWHP1),IPETA1,TEMPA)
TEMPA(1)=ISUBO
TEMPA(2)=ISUBN
TEMPA(3)=NUMPATH
TEMPA(4)=DALT
TEMPA(5)=AALT
TEMPA(6)=ETASEP
CALL
CALL PAKSPL(IONE,ISIX,IETA2(ISUBN,IWHP1),IPETA2,TEMPA)
C**** UPDATE ACTIVE SCHEDULE
25000
IF(ITRYSH.EQ.0)
TOP OF NEXT PAGE
GO TO 25010
M17 PG.102
PAGE 100
BOTTOM OF PRECEDING PAGE
CALL
CALL UPKSPL(IFOUR,IFIVE,NNPATH(KPATHS),IPNPATH,TEMPA)
LLL=NNA(IAC)
LLLP3=LLL+3
IASKED(IAC,LLLP3)=IASKED(IAC,LLL)
ASKSPD(IAC,LLLP3) = 999.
IF(NHOLDP(IAC).LE. 1)
GO TO 25005
M14 PG.101
LLLM1 = LLL-1
LLLP2 = LLL+2
IASKED(IAC,LLLP2) = IASKED(IAC,LLLM1)
ASKSPD(IAC,LLLP2) = 999.
ITEMP = TEMPA(1)
CALL
CALL UPKSPL(IONE, IEIGHT, ISKED(ITEMP), IPSKED, TEMPA3)
TEMPA3(5) = TEMPA3(6)
TEMPA3(6) = TEMPA3(7)
TEMPA3(7) = TEMPA3(8)
CALL
CALL UPKSPL(ITWO, ITWO, IASKED(IAC,LLLP2), IPASKED, TEMP)
LLLP1 = LLL+1
TEMPA3(2) = AMOD(TEMP+180., 360.)
CALL
CALL PAKSPL(IONE, ISEVEN, IASKED(IAC,LLLP1), IPASKED, TEMPA3)
ASKSPD(IAC,LLLP1) = 999.

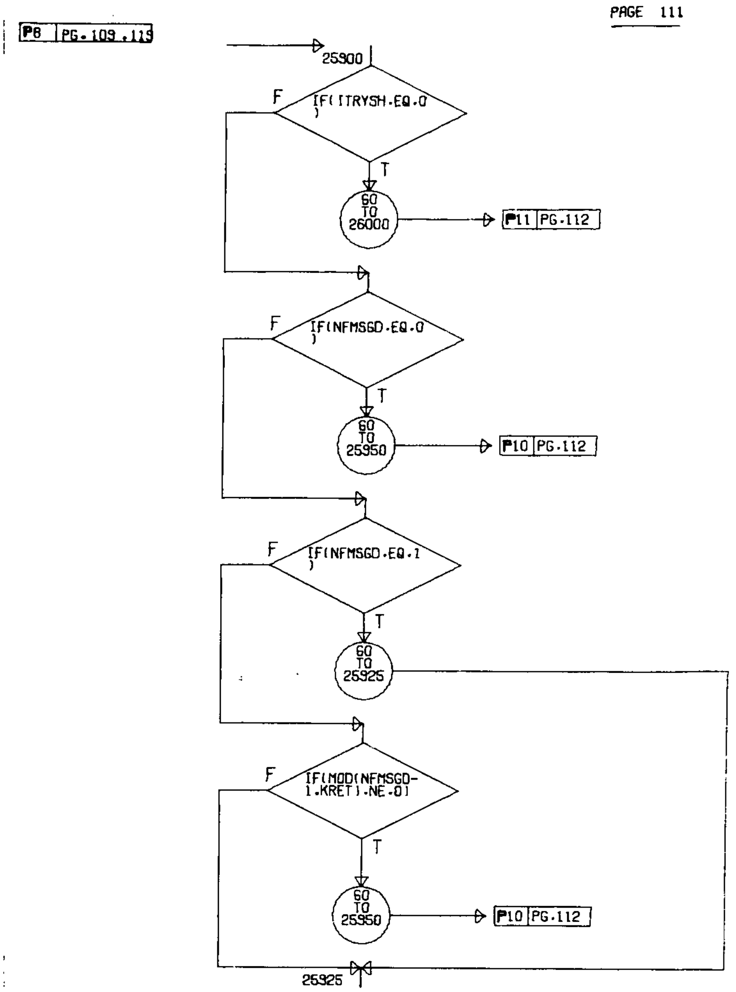


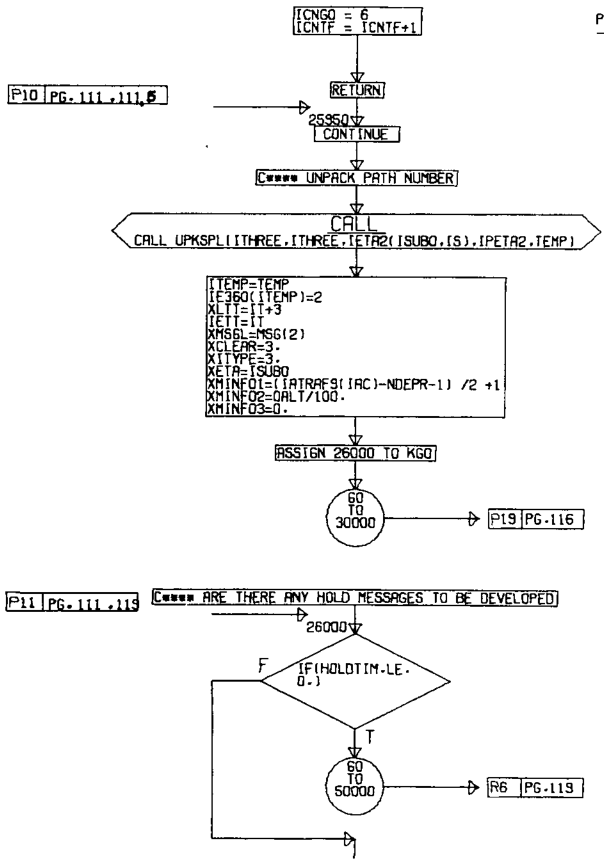
PAGE 101
GO TO 26009
M16 PG.102
M14 PG.100
26005 CONTINUE
ITEMP=TEMPA(1)
CALL
CALL UPKSPL(IONE,IEIGHT,ISKED(ITEMP),IPSKED,TEMPA3)
HDG = 0.
CALL
CALL UPKSPL(IONE, IONE, IASKED(IAC,1), IPASKED, TEMP)
IF(TEMP .NE. 9 .)
F
T
GO TO 26008
CALL
CALL UPKUNI(ICLASS, ITEN, IPFTR, 10., TEMP)
CALL
CALL UPKSPL(ISEVEN, ISEVEN, IASKED(IAC,1), IPASKED, TEMPS)
HDG = (TEMPS-T(IAC))*TEMP
26008 CONTINUE
TEMPA3(2) = ATRAF6(IAC)+180.+HDG
TEMPA3(2)=AMOD(TEMPA3(2),360.)
LLL=LLL+1
TEMPA3(5)=TEMPA3(6)
TEMPA3(6)=TEMPA3(7)
TEMPA3(7)=TEMPA3(8)
ASKSPD(IAC,LLL)=999.
PAGE 102
CALL
CALL PAKSPL(IONE,ISEVEN,IASKED(IAC,LLL),IPASKED,TEMPA3)
LLL=LLL+1
ITEMP=TEMPA(2)
CALL
CALL UPKSPL(IONE,IEIGHT,ISKED(ITEMP),IPSKED,TEMPA3)
TEMPA3(2) = AMOD(ATRAF6(IAC)+HDG, 360.)
TEMPA3(5)=TEMPA3(6)
TEMPA3(6)=TEMPA3(7)
TEMPA3(7)=TEMPA3(8)
ASKSPD(IAC,LLL)=999.
CALL
CALL PAKSPL(IONE,ISEVEN,IASKED(IAC,LLL),IPASKED,TEMPA3)
16 PG. 101
26009 CONTINUE
NNA(IAC)=LLLP3
NHOLDP(IAC) = NNA(IAC)
17 PG. 100
26010 JOE=NNA(IAC)
IATRAF(IAC,26)=ICACT2
IF(IVECT.EQ.0.OR.NSEGM.EQ.0)
F
T
GO TO 26400
N10 PG.107
DO 26300 K=1,NSEGM
N7 PG.105
KP3=K+3
PAGE 103
C**** UNPACK SUBSCRIPT OF NEXT SEGMENT TO BE TRANSFERRED FROM MASTER
C**** SCHEDULE TO ACTIVE
CALL
CALL UPKSPL(KP3,KP3,NNPATH(IVECT),IPNPATH,TEMP)
IWHCH=TEMP
C**** UNPACK ELEMENTS OF THE SEGMENT
CALL
CALL UPKSPL(IONE,IEIGHT,ISKED(IWHCH),IPSKED,TEMPA)
ITEMP=TEMPA(6)
JOE=JOE+1
C**** IS THE SPEED IN THE SCHEDULE MISSING OR EQUAL TO THE DESIRED SPD
IF(ITEMP.EQ.0 .OR.(ITEMP.EQ.IDSPD .AND. ISPDOP.GT.0))
F
T
GO TO 25025
M19 PG.104
DO 25015 J=1,4
25015 JTEMPA(J) = IPERFS(J,ITEMP)
C**** UNPACK AND INSERT SPEED DICTATED BY THE SCHEDULE
CALL
CALL UPKUNI(ICLASS,ISIX,JTEMPA,1.,TEMP)
ASKSPD(IAC ,JOE)=TEMP/3600.
PAGE 104
GO TO 25036
C**** SET THE SPEED TO MISSING BECAUSE IT WILL HOLD CONSTANT UNTIL
C**** CHANGED BY A MESSAGE
M18 PG.103
25025
IF(JOE.EQ.1)
F
T
GO TO 25030
ASKSPD(IAC,JOE)=999.
GO TO 25036
25030
ASKSPD(IAC,JOE)=ATRAF6(IAC)
25036
ITEMP=TEMPA(4)
IF(ITEMP.EQ.0) 25050,25076
T
F
N3 PG.105
C**** HOLD ALTITUDE
25050
TEMPA(4)=0.

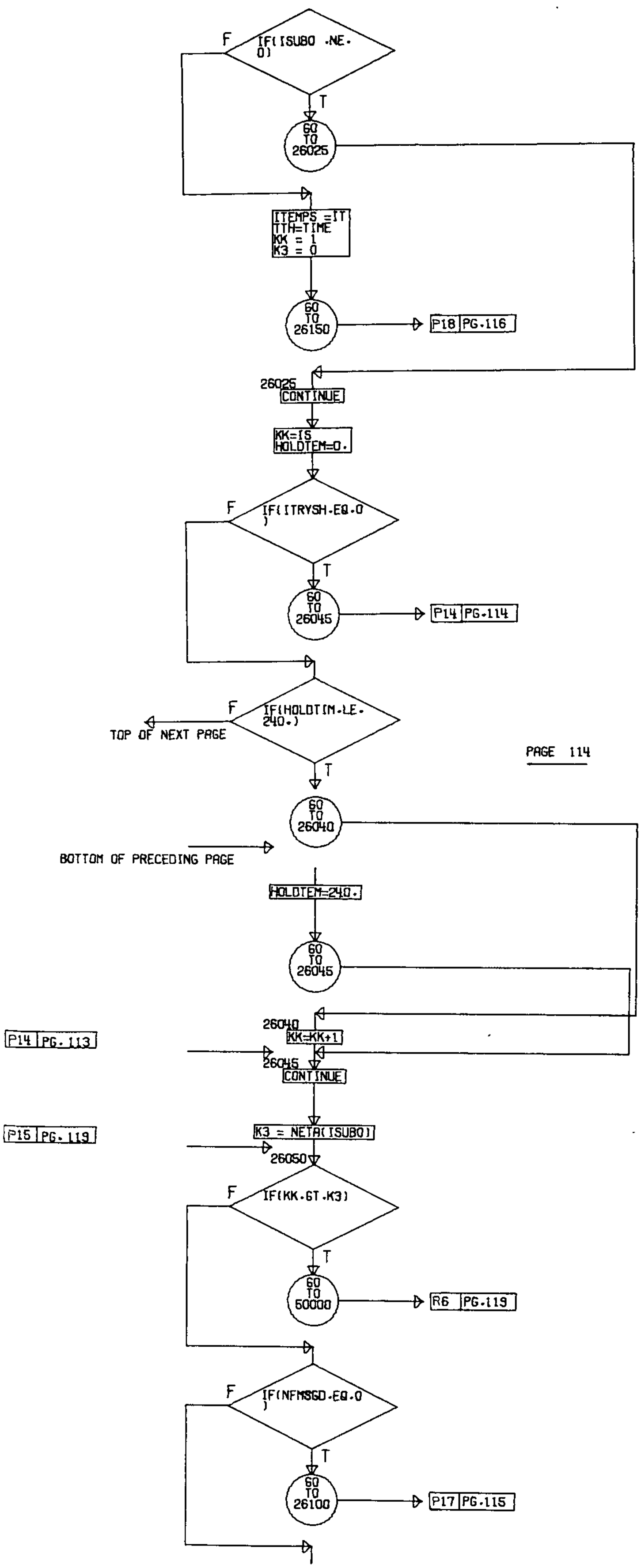

GO TO 25200
N3 PG. 104
25076
IF(ITEMP.EQ.31) 25100,25126
T
F
C**** INSERT DESIRED ALTITUDE
25100
TEMPA(4)=AALT
GO TO 25200
C**** INSERT SCHEDULED ALTITUDE
25126
TEMPA(4)=ALTSKD(ITEMP)
C**** MOVE REST OF SEGMENT ELEMENTS TO BE PACKED INTO ASKED
25200
TEMPA(5)=TEMPA(6)
TEMPA(6)=TEMPA(7)
TEMPA(7)=TEMPA(8)
N7 PG. 102
C**** PACK SEGMENT
25300
CALL
CALL PAKSPL(IONE.ISEVEN.(ASKED(IAC .JOE).IPASKED.TEMPA)
NNA(IAC)=JOE
C**** SAVE PATH INFO IN ATRAF
TEMPA(1)=OALT
TEMPA(2)=AALT
TEMPA(3)=NUMPATH
TEMPA(4)=ISUBO
TEMPA(5)=ISUBN
PAGE 106
CROSSHAIR NO. 1
IF(ITYPCL.EQ.6 .OR. ITYPCL.EQ.7 .OR. ITYPCL.EQ.11) 25350, 25376
T
F
25350
CALL
CALL PAKSPL(IONE. IFIVE. IATRF21(IAC). IPAPATH. TEMPA)
GO TO 25400
25376
CONTINUE
CALL
CALL PAKSPL(IONE.IFIVE.(ATRF23(IAC).IPAPATH.TEMPA)
PAGE 1
N10 PG.102
C**** MESSAGES
25400
GO TO (25500,25600,25700,50000,25700,25700,25800,50000,50000,50000,25700) ,ITYPCL
P2 PG.107
P3 PG.107
P4 PG.108
A6 PG.119
P4 PG.108
P4 PG.108
P5 PG.109
A6 PG.119
A6 PG.119
A6 PG.119
P4 PG.108
P2 PG. 107
C**** ARRIVAL CLEARANCE MESSAGE
25500
IETT=IT
XLTT=IT+1
XMSGL=MSG(1)
XCLEAR=NSEGM
XITYPE=2.
XETA=ISUBN
XMINFO1=(IATRAF9(IAC)-NDEPR-1)/2+1
XMINFO2=AALT/100.
XMINFO3=0.
ASSIGN 25800 TO KGO
GO TO 30000
P19 PG.116
P3 PG. 107
C**** DEPARTURE CLEARANCE
25600
IETT=IT
XLTT=IT+1
XMSGL=MSG(7)
XMINFO3=0.
XCLEAR=NSEGM
XITYPE=1.
XETA=ISUBN
XMINFO1=AALT/100.
XMINFO2=IATRF12(IAC)
ASSIGN 50000 TO KGO
PAGE 108
GO TO 30000
P19 PG.116
C**** NORMAL FLIGHT WITH GENERAL CONFLICT CHECK
P4 PG. 107, 107, 107, 107
25700
IF(IVECT.EQ.0)
F
T
GO TO 25800
P5 PG.109
ITEMPS=TETAO/DELT
IF(AMOD(TETAO, DELT).GT.0.)
F
T
ITEMPS = ITEMPS+1
XLTT=ITEMPS
IETT=ITEMPS-60/IDELT
XMSGL=MSG(5)
XCLEAR=NSEGM
XITYPE=0.
XETA=ISUBN
XMINFO1=0.
XMINFO2=0.
XMINFO3=0.
ASSIGN 25800 TO KGO
GO TO 30000
P19 PG.116
C**** SPEED CONTROL MESSAGE

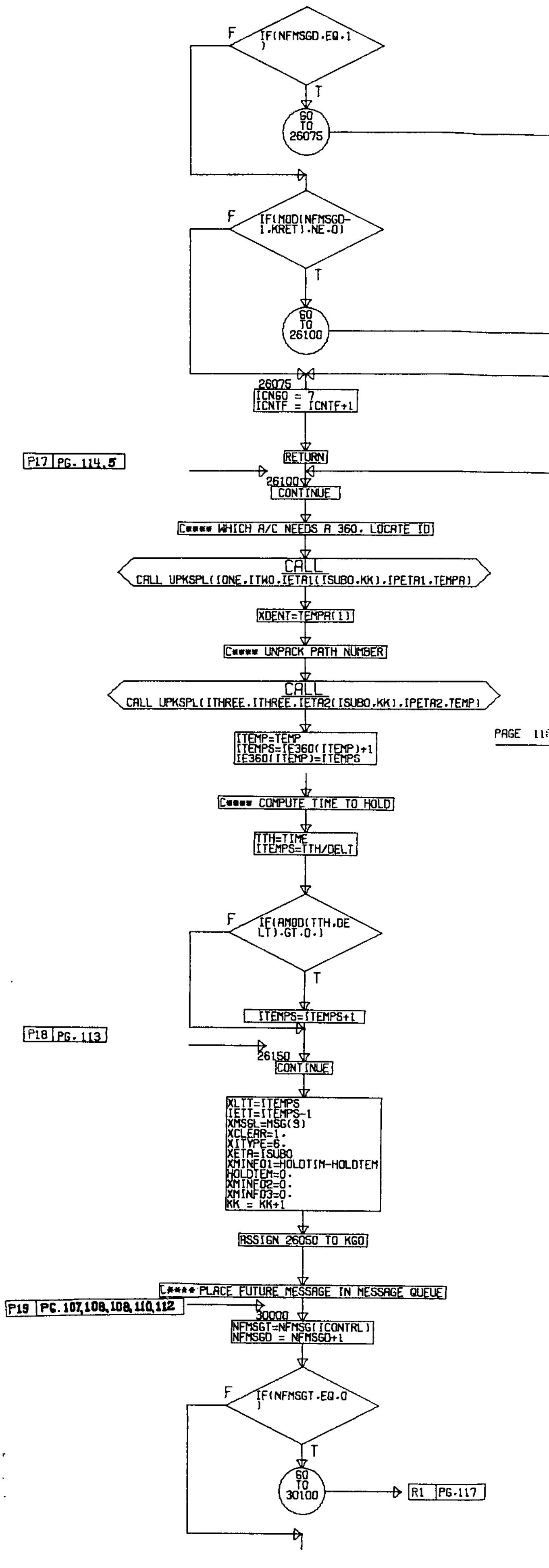

PAGE 109
PG. 107 .108 113
25800
IF(ISPOOP .EQ.0 .AND. ISPMSG. EQ.0)
F
T
GO TO 25900
P8 PG.111
IF(NFMSGD.EQ.0)
GO TO 25850
P7 PG.110
IF(NFMSGD.EQ.1)
GO TO 25825
IF(MOD(NFMSGD- 1.KRET).NE.0)
GO TO 25850
P7 PG.110
25825


ICNGO = 5
ICNTF = ICNTF+1
PAGE 110
RETURN
PG. 109 .109.5
25850
CONTINUE
IF(IDSPD.NE.0 . AND. INDSPD.EQ. 0 .AND. DSPD.LT .0SPD)
F
T
TX = TX3
TEMP = TX+TETA0+CETA
ITEMPS=TEMP/DELT
IF(AMOD(TEMP.D ELT).GT.0.)
ITEMPS=ITEMPS+1
XLTT=ITEMPS
IETT=ITEMPS-MSG(4)
XMSGL=MSG(4)
XCLEAR=0.
XITYPE=8.
XETA=ISUBN
XMINFO1=DSPD*3600.
XMINFO2=0.
XMINFO3=0.
ASSIGN 25900 TO KGO
GO TO 30000
P19 PG.116
C**** STANDARD HOLD


PAGE 111
P8 PG. 109 .113
25900
IF(ITRYSH.EQ.0)
F
T
GO TO 26000
P11 PG.112
IF(NFMSGD.EQ.0)
GO TO 25950
P10 PG.112
IF(NFMSGD.EQ.1)
GO TO 25925
IF(MOD(NFMSGD- 1.KRET).NE.0)
GO TO 25950
P10 PG.112
25925

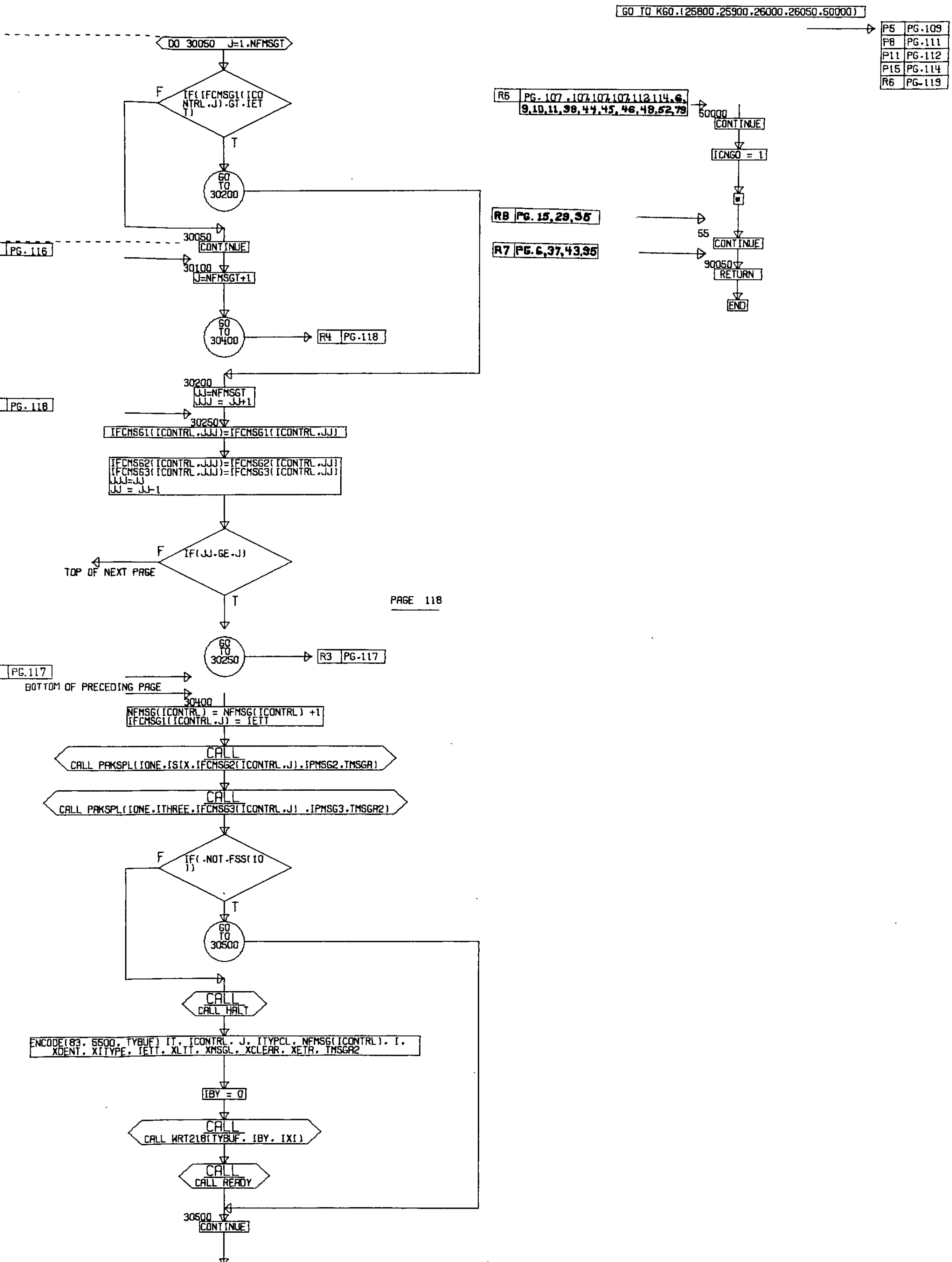

ICNGO = 6
ICNTF = ICNTF+1
PAGE 112
RETURN
P10 PG. 111 .111.5
25950
CONTINUE
C**** UNPACK PATH NUMBER
CALL
CALL UPKSPL(ITHREE, ITHREE, IETA2(ISUBO, IS), IPETA2, TEMP)
ITEMP=TEMP
IE360(ITEMP)=2
XLTT=IT+3
IETT=IT
XMSGL=MSG(2)
XCLEAR=3.
XITYPE=3.
XETA=ISUBO
XMINFO1=(IATRAFS(IAC)-NDEPR-1) /2 +1
XMINFO2=QALT/100.
XMINFO3=0.
ASSIGN 26000 TO KGO
GO TO 30000
P19 PG.116
P11 PG. 111 .113
C**** ARE THERE ANY HOLD MESSAGES TO BE DEVELOPED
26000
IF(HOLDTIM.LE. 0.)
F
T
GO TO 50000
R6 PG.119


IF(ISUBO .NE. 0)
F
T
GO TO 26025
ITEMPS =IT
TTH=TIME
KK = 1
K3 = 0
GO TO 26150
P18 PG.116
26025
CONTINUE
KK=IS
HOLDTEM=0.
IF(ITRYSH.EQ.0)
F
T
GO TO 26045
P14 PG.114
IF(HOLDTIM.LE. 240.)
F
TOP OF NEXT PAGE
T
PAGE 114
GO TO 26040
BOTTOM OF PRECEDING PAGE
HOLDTEM=240.
GO TO 26045
26040
KK=KK+1
P14 PG.113
26045
CONTINUE
P15 PG.119
K3 = NETA(ISUBO)
26050
IF(KK.GT.K3)
F
T
GO TO 60000
R6 PG.119
IF(NFMSGD.EQ.0)
F
T
GO TO 26100
P17 PG.115


IF(NFMSGD.EQ.1)
F
T
GO TO 26075
IF(MOD(NFMSGD-1,KRET).NE.0)
F
T
GO TO 26100
26075
ICNGO = 7
ICNTF = ICNTF+1
RETURN
P17 PG.114.5
26100
CONTINUE
C**** WHICH A/C NEEDS A 360. LOCATE ID
CALL
CALL UPKSPL(IONE,ITWO,IETA1(ISUBO,KK),IPETA1,TEMPA)
XDENT=TEMPA(1)
C**** UNPACK PATH NUMBER
CALL
CALL UPKSPL(ITHREE,ITHREE,IETA2(ISUBO,KK),IPETA2,TEMP)
PAGE 11
ITEMP=TEMP
ITEMPS=IE360(ITEMP)+1
IE360(ITEMP)=ITEMPS
C**** COMPUTE TIME TO HOLD
TTH=TIME
ITEMPS=TTH/DELT
IF(AMOD(TTH,DELT).GT.0.)
F
T
ITEMPS=ITEMPS+1
P18 PG.113
26150
CONTINUE
XLTT=ITEMPS
IETT=ITEMPS-1
XMSGL=MSG(9)
XCLEAR=1.
XITYPE=6.
XETA=ISUBO
XMINFO1=HOLDTIM-HOLDTEM
HOLDTEM=0.
XMINFO2=0.
XMINFO3=0.
KK = KK+1
ASSIGN 26050 TO KGO
C**** PLACE FUTURE MESSAGE IN MESSAGE QUEUE
P19 PG.107,108,108,110,112
30000
NFMSGT=NFMSG(ICONTRL)
NFMSGD = NFMSGD+1
IF(NFMSGT.EQ.0)
F
T
GO TO 30100
R1 PG.117

PAGE 117

PAGE 119



PAGE 1

```
SUBROUTINE ETACOMP

LOGICAL LOISI, LDISO

COMMON / REALTIM/ ADC(32),DAC(64),LDISI(108),LOISO(196),
NOPER,NHOLD,NRESET,NTERM,NPRINT,NREAD

COMMON/TABLES/
    INTEG(80),      TABLE(076),      LOGIC(20)

LOGICAL LOGIC

*****
* * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN
*****

COMMON/MAJOR1/
     ACLS(50),        ALTOPT(3,7,2),  ALTSKD(9),
     ARRMNA,          ARRMXA,         BOUND,
     CGLIDE,          CILS,           CRA20,
     CRA30,           CRB20,          CRB30,
     DGRTD,           DTR,            EHOLDD,
     ENROTA(2),       FTNM,           G,
     GLIDE(2),        HOLDSEP,        IARRN,
     IARRS,           IART(11),       IBF(110),
     ICHKPT(40),      ICNTRA1(85),    ICNTRA2(85),
     IDELT,           IDEPN,          IDEPS,
     IDROUTE(10),     IHPATHE,        IHPATHS,
     INAV(10,5),      IOPTN(55),      IPAPATH(4,6),
     IPASKED(4,7),    IPCHKPT(4,6),   IPCNTR1(4,6),
     IPCNTR2(4,2),    IPDRT(4,5),     IPERFR(3,7),
     IPERFS(4,8),     IPERFT(3),      IPETA1(4,4),
     IPETA2(4,6),     IPFTR(3),       IPMSG2(4,6),
     IPMSG3(4,3),     IPNPATH(4,7),   IPNSITE(4,4),
     IPOPL(2),        IPOPPA,         IPOPR(2),
     IPOPTN(4,9),     IPSKED(4,8),    IPTAS(4,5)

COMMON/MAJOR2/
     IPTRAF1(4,4),    IPTRAF2(4,9),   ISDST(17),
     ISKDPU,          ISKED(86),      ISTARTA(21),
     ITANG(2,2),      ITRAF1(102),    ITRAF2(102),
     MAXK,            MF,
     NALTSKD,         NCP,            NDEPR,
     NNAV(5),         NNPATH(50),     NOUT(20),
     NSPDW,           OUT(26),        PI,
     PPMX(3),         PPMY(3),        RCP(2),
     RTD,             RWYX(4,2),      RWYY(4,2),
     RWYZ(2),         SHOLDD,         SPDSTD(9,2),
     SQX(6),          SQY(6),         TRIX(5),
     TRIY(5),         XA(2),          XC(2),
     XDILS(2),        XDIST(2),       XFINAL(2),
     XRPOS(2),        YA(2),          YC(2),
     YRPOS(2),        RAD(3),         SIG(3),        IARG
   . IETCOOR(36),     IPETC(4,2),     IACFPER(3)

*****

COMMON/TOES/ IONE,ITWO,ITHREE,IFOUR,IFIVE,ISIX,ISEVEN,IEIGHT,ININE
.ITEN
```

PAGE 2

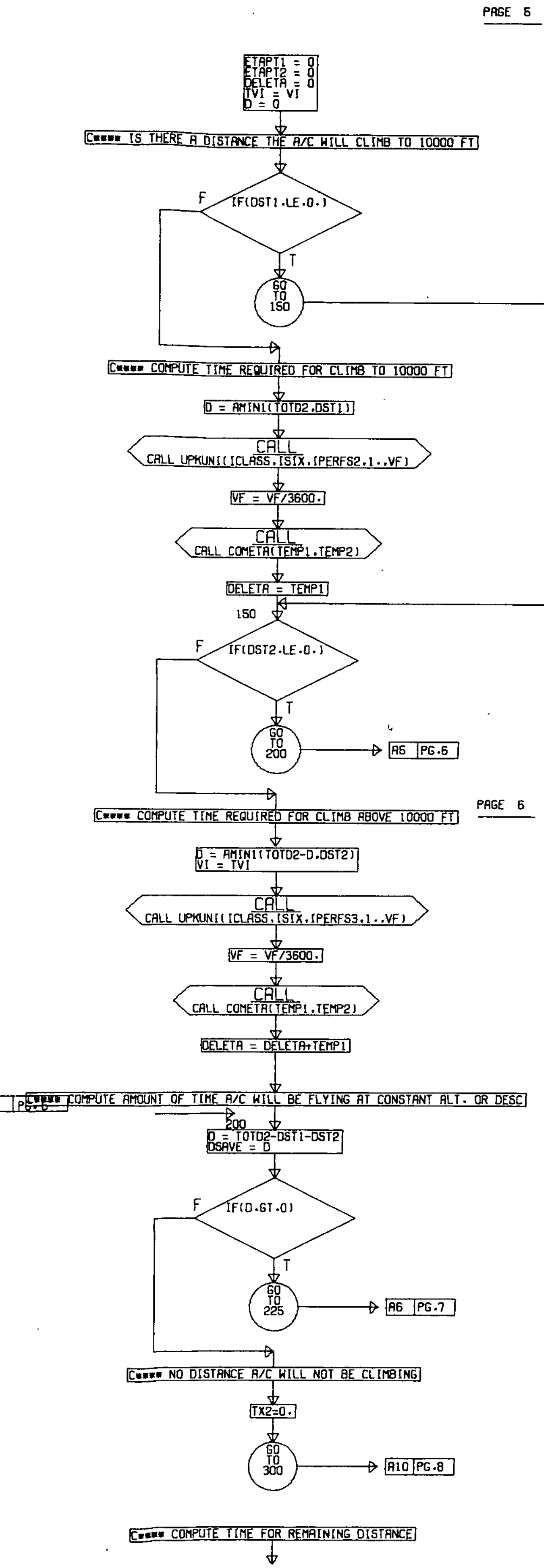
```
*****
* * * REAL-TIME MAIN COMMON IN OVERLAY(4,0) AND ALL SECONDARY OVERLAYS A
* * *     PRIMARY SUBROUTINES OF (4,0)
*****

COMMON/RTMAIN1/
    ASKSPD(36,11),ASPACE1(4,9),ASPACE2(4,9),ASPACE3(4,9),
     CETA,            CONFLCT,        DELH(35),
     HDBEG(36),       HDOWN,          IA(2),
     IRC,             IRCTN1(35),     IACTN2(36),
  IACTN3(36), IASKED(36,11), IATAS(36,2),ICNGO,
     IATRAF(36,26),   ICA,            ICACT,
     ICLASS,          ICMSG1(8,5),    ICMSG2(8,5),
     ICMSG3(8,5),     ICOMM(10),      ID(2),
     IDELAY,          IENROT(2,20),   IETAS(36,2),
     IETA1(36,10),    IETA2(36,10),   IE360(60),
     IFCMSG1(8,25),   IFCMSG2(8,26),  IFCMSG3(8,26),
     IHOWN,           IOBETA,         IOEETA,
     IOPATH,          IPASSAC,        ISEQ,
     ISEQUE(3,16),    ISW4,           ITOQUE(2,20),
     ITRY,            IWHICH,         KNTP,
     L,               LEMG(6),        NASPACE(4),
     NCMSG(10),       NEMG,           NETA(36),
     NFMSG(10),       NHOLDP(35),     NNA(35),
     NSEQ,            OLDR(36)

COMMON/RTMAIN2/
     OTHAA,           OTHETA,         OTHOA,
     OTHSETA,         RESTOR(35),     ROLL,
     T(35),           TETAE,          TETAL,
     TETAO,           TIME,           V1,
     V2,              XOWN,           XPL(35),
     YOWN,            YPL(35),        ZOWN
   . R9LX(2),         R9LY(2),        R9RX(2),       R9RY(2),
     R15X(2),         R16Y(2),        R21X(2),       R21Y(2),
     XNREF(4),        YNREF(4)

*****

COMMON/B1/ VI, VF, D, ACCEL, TVI

DIMENSION
    IOPTION(9),     OPTION(9),     IPERFS2(4),     IPERFS3(4)

EQUIVALENCE
    (IPERFS(1,2), IPERFS2(1)),     (IPERFS(1,3), IPERFS3(1))
```

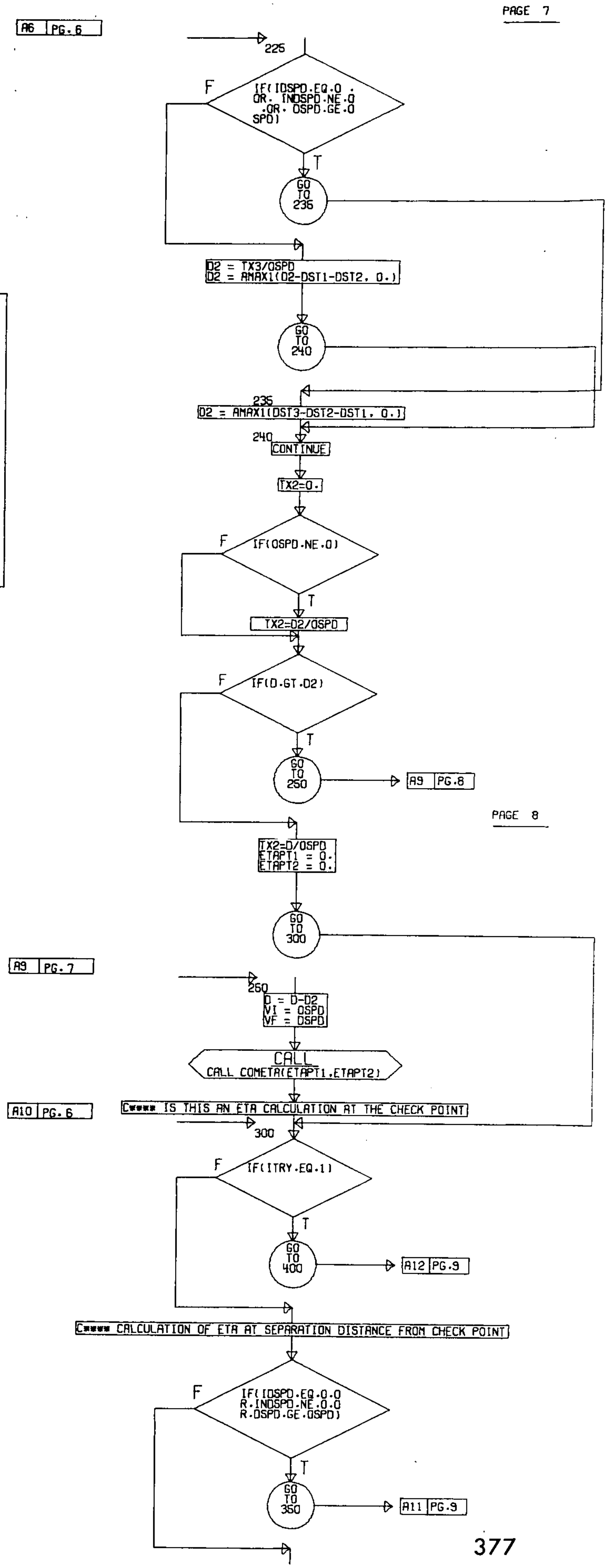

PAGE 3

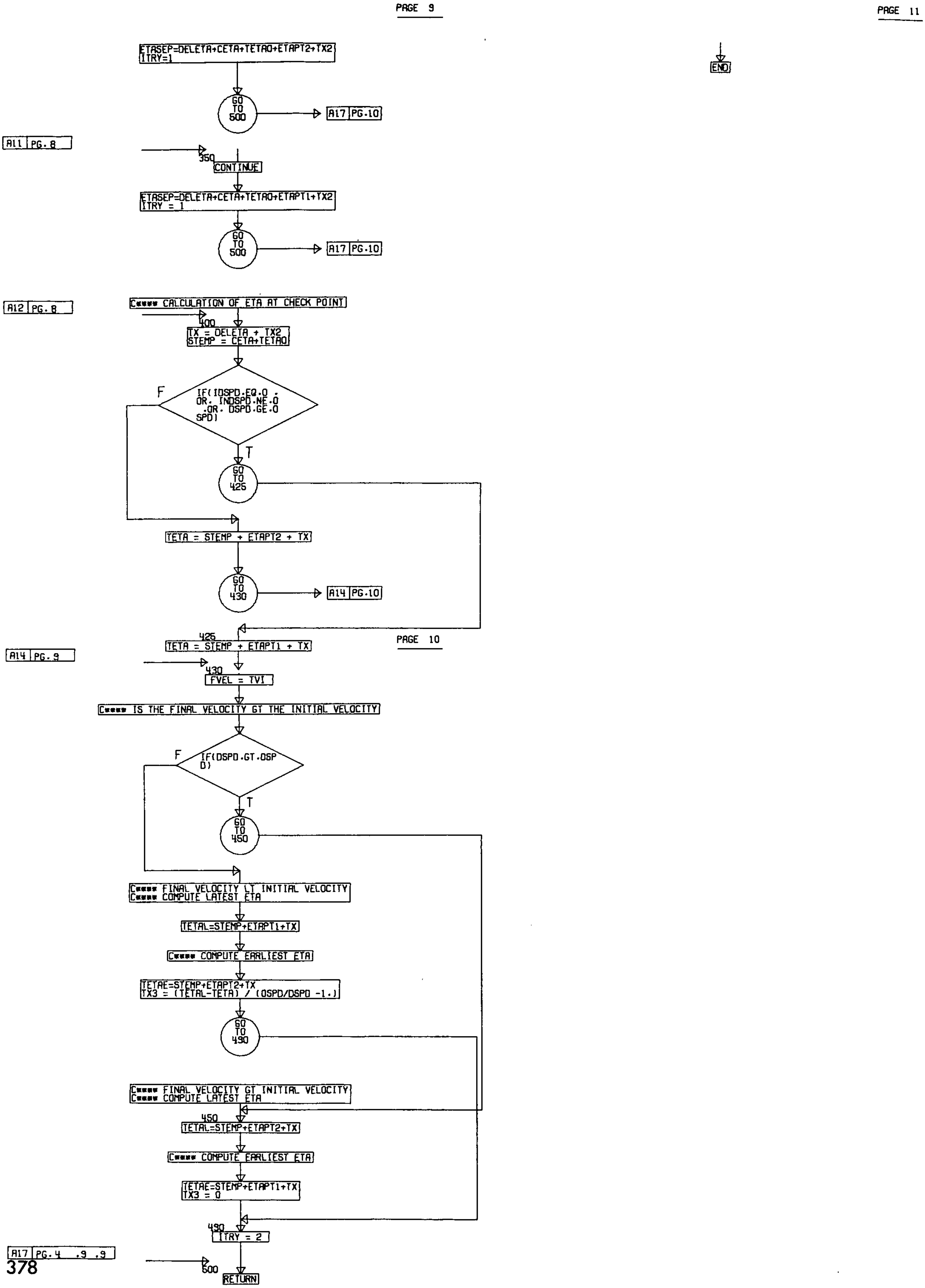
```
EQUIVALENCE
    (RESTOR(01), TETA),            (RESTOR(02), NUMPATH),
    (RESTOR(03), OALT),            (RESTOR(04), AALT),
    (RESTOR(05), ETASEP),          (RESTOR(06), FVEL),
    (RESTOR(07), DSPD),            (RESTOR(08), TX),
    (RESTOR(09), TOTD),            (RESTOR(10), NSEGM),
    (RESTOR(11), DST1),            (RESTOR(12), DST2),
    (RESTOR(13), DST3),            (RESTOR(14), ISUBCK),
    (RESTOR(15), ISUBOP),          (RESTOR(16), ICACT2),
    (RESTOR(17), STEMP),           (RESTOR(18), ISUBN),
    (RESTOR(19), ISUBO),           (RESTOR(20), SEP),
    (RESTOR(21), XLA),             (RESTOR(22), XLB),
    (RESTOR(23), IASPCON),
    (RESTOR(24), IDALT, IOPTION(1), OPTION(1)),
    (RESTOR(25), IDSPD),           (RESTOR(26), IVECT),
    (RESTOR(27), IVECOP),          (RESTOR(28), IDVERG2),
    (RESTOR(29), IALTOP),          (RESTOR(30), ISPDOP),
    (RESTOR(31), ISHOP),           (RESTOR(32), IEHOP)
  . (RESTOR(33), OSPD)
  . (RESTOR(34), TX3),             (RESTOR(35), INDSPD)

VF = DSPD
VI = OSPD
TOTD2 = TOTD

IF(ITRY.EQ.1)      T: GO TO 100  --> A3 PG.4
                   F: (continue)

C**** ETA COMPUTATION WILL BE FOR POINT AT SEPARRAION DISTANCE FROM
C****    CHECK POINT
CR*** IS THERE A COMMON PATH CALCULATION

RO = 0

IF(XLA.EQ.0.AND.XLB.EQ.0.)      T: GO TO 90  --> A2 PG.4
                                F: (continue)
```

PAGE 4

```
C**** COMMON PATH CALCULATION

IF(V1.GT.V2)      T: GO TO 50
                  F: (continue)

RO = (V2/V1-1.)*XLA

GO TO 90

A2 PG.3
50
RO = (1.-V2/V1)*XLB
90
TOTD2=TOTD-RO-SEP

IF(TOTD2.GT.0)      T: GO TO 100
                    F: (continue)

ETASEP = 0.
ITRY = 1

GO TO 500  --> A17 PG.10

A3 PG.3
C**** INITIALIZE TIMES
100
```


PAGE 5
ETAPT1 = 0
ETAPT2 = 0
DELETA = 0
TVI = VI
D = 0
C**** IS THERE A DISTANCE THE A/C WILL CLIMB TO 10000 FT
IF(DST1.LE.0.)
F
T
GO TO 150
C**** COMPUTE TIME REQUIRED FOR CLIMB TO 10000 FT
D = AMIN1(TOTD2,DST1)
CALL
CALL UPKUNI(ICLASS,ISIX,IPERFS2,1.,VF)
VF = VF/3600.
CALL
CALL COMETA(TEMP1,TEMP2)
DELETA = TEMP1
150
IF(DST2.LE.0.)
F
T
GO TO 200
A5 PG.6
PAGE 6
C**** COMPUTE TIME REQUIRED FOR CLIMB ABOVE 10000 FT
D = AMIN1(TOTD2-D,DST2)
VI = TVI
CALL
CALL UPKUNI(ICLASS,ISIX,IPERFS3,1.,VF)
VF = VF/3600.
CALL
CALL COMETA(TEMP1,TEMP2)
DELETA = DELETA+TEMP1
C**** COMPUTE AMOUNT OF TIME A/C WILL BE FLYING AT CONSTANT ALT. OR DESC
PG.6
200
D = TOTD2-DST1-DST2
DSAVE = D
IF(D.GT.0)
F
T
GO TO 225
A6 PG.7
C**** NO DISTANCE A/C WILL NOT BE CLIMBING
TX2=0.
GO TO 300
A10 PG.8
C**** COMPUTE TIME FOR REMAINING DISTANCE


A6 PG.6
PAGE 7
225
IF(IDSPD.EQ.0.OR.INDSPD.NE.0.OR.OSPD.GE.OSPD)
F
T
GO TO 235
D2 = TX3/OSPD
D2 = AMAX1(D2-DST1-DST2, 0.)
GO TO 240
235
D2 = AMAX1(DST3-DST2-DST1, 0.)
240
CONTINUE
TX2=0.
IF(OSPD.NE.0)
F
T
TX2=D2/OSPD
IF(D.GT.D2)
F
T
GO TO 250
A9 PG.8
PAGE 8
TX2=D/OSPD
ETAPT1 = 0.
ETAPT2 = 0.
GO TO 300
A9 PG.7
250
D = D-D2
VI = OSPD
VF = OSPD
CALL
CALL COMETA(ETAPT1,ETAPT2)
A10 PG.6
C**** IS THIS AN ETA CALCULATION AT THE CHECK POINT
300
IF(ITRY.EQ.1)
F
T
GO TO 400
A12 PG.9
C**** CALCULATION OF ETA AT SEPARATION DISTANCE FROM CHECK POINT
IF(IDSPD.EQ.0.OR.INDSPD.NE.0.OR.OSPD.GE.OSPD)
F
T
GO TO 350
A11 PG.9


PAGE 9
PAGE 11
ETASEP=DELETA+CETA+TETAO+ETAPT2+TX2
ITRY=1
GO TO 500
A17 PG.10
END
A11 PG. 8
350 CONTINUE
ETASEP=DELETA+CETA+TETAO+ETAPT1+TX2
ITRY = 1
GO TO 500
A17 PG.10
A12 PG. 8
C**** CALCULATION OF ETA AT CHECK POINT
400
TX = DELETA + TX2
STEMP = CETA+TETAO
IF(IDSPD.EQ.0 .OR. INDSPD.NE.0 .OR. DSPD.GE.0 SPD)
F
T
GO TO 425
TETA = STEMP + ETAPT2 + TX
GO TO 430
A14 PG.10
425
TETA = STEMP + ETAPT1 + TX
PAGE 10
A14 PG. 9
430 FVEL = TVI
C**** IS THE FINAL VELOCITY GT THE INITIAL VELOCITY
IF(DSPD.GT.OSPD)
F
T
GO TO 450
C**** FINAL VELOCITY LT INITIAL VELOCITY
C**** COMPUTE LATEST ETA
TETAL=STEMP+ETAPT1+TX
C**** COMPUTE EARLIEST ETA
TETAE=STEMP+ETAPT2+TX
TX3 = (TETAL-TETA) / (OSPD/DSPD -1.)
GO TO 490
C**** FINAL VELOCITY GT INITIAL VELOCITY
C**** COMPUTE LATEST ETA
450
TETAL=STEMP+ETAPT2+TX
C**** COMPUTE EARLIEST ETA
TETAE=STEMP+ETAPT1+TX
TX3 = 0
490 ITRY = 2
A17 PG. 4 .9 .9
500 RETURN

```
SUBROUTINE CONCHK

LOGICAL LDISI. LDISO

COMMON / REALTIM/ ADC(32).DAC(64).LDISI(108).LDISO(196).
NOPER.NHOLD.NRESET.NTERM.NPRINT.NREAD

COMMON/TABLES/
   INTEG(80).      TABLE(075).      LOGIC(20)

LOGICAL LOGIC

*****
* * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN
*****

COMMON/MAJOR1/
     ACLS(50).         ALTOPT(3.7.2). ALTSKD(9).
     ARRMNA.           ARRMXA.        BOUND.
     CGLIDE.           CILS.          CRA20.
     CRA30.            CRB20.         CRB30.
     DGRTD.            DTR.           EHOLDD.
     ENROTA(2).        FTNM.          G.
     GLIDE(2).         HOLDSEP.       IARRN.
     IARRS.            IART(11).      IBF(110).
     ICHKPT(40).       ICNTRA1(85).   ICNTRA2(85).
     IDELT.            IDEPN.         IDEPS.
     IDROUTE(10).      IHPATHE.       IHPATHS.
     INAV(10.5).       IOPTN(55).     IPAPATH(4.5).
     IPASKED(4.7).     IPCHKPT(4.6).  IPCNTR1(4.6).
     IPCNTR2(4.2).     IPORT(4.5).    IPERFR(3.7).
     IPERFS(4.8).      IPERFT(3).     IPETA1(4.4).
     IPETA2(4.6).      IPFTR(3).      IPMSG2(4.6).
     IPMSG3(4.3).      IPNPATH(4.7).  IPNSITE(4.4).
     IPOPL(2).         IPOPPA.        IPOPR(2).
     IPOPTN(4.9).      IPSKED(4.8).   IPTAS(4.5)

COMMON/MAJOR2/
     IPTRAF1(4.4).     IPTRAF2(4.9).  ISDST(17).
     ISKDPU.           ISKED(85).     ISTARTA(21).
     ITANG(2.2).       ITRAF1(102).   ITRAF2(102).
     MAXK.             MF.
     NALTSKD.          NCP.           NDEPR.
     NNAV(5).          NNPATH(50).    NOUT(20).
     NSPDW.            OUT(26).       PI.
     PPMX(3).          PPMY(3).       RCP(2).
     RTD.              RWYX(4.2).     RWYY(4.2).
     RWYZ(2).          SHOLDD.        SPDSTD(9.2).
     SQX(6).           SQY(6).        TRIX(5).
     TRIY(5).          XA(2).         XC(2).
     XDILS(2).         XDIST(2).      XFINAL(2).
     XRPOS(2).         YA(2).         YC(2).
     YRPOS(2).         RAD(3).        SIG(3).        IARG
   . IETCOOR(36).      IPETC(4.2).    IACFPER(3)

*****

COMMON/TOES/ IONE.ITWO.ITHREE.IFOUR.IFIVE.ISIX.ISEVEN.IEIGHT.ININE
.ITEN
```



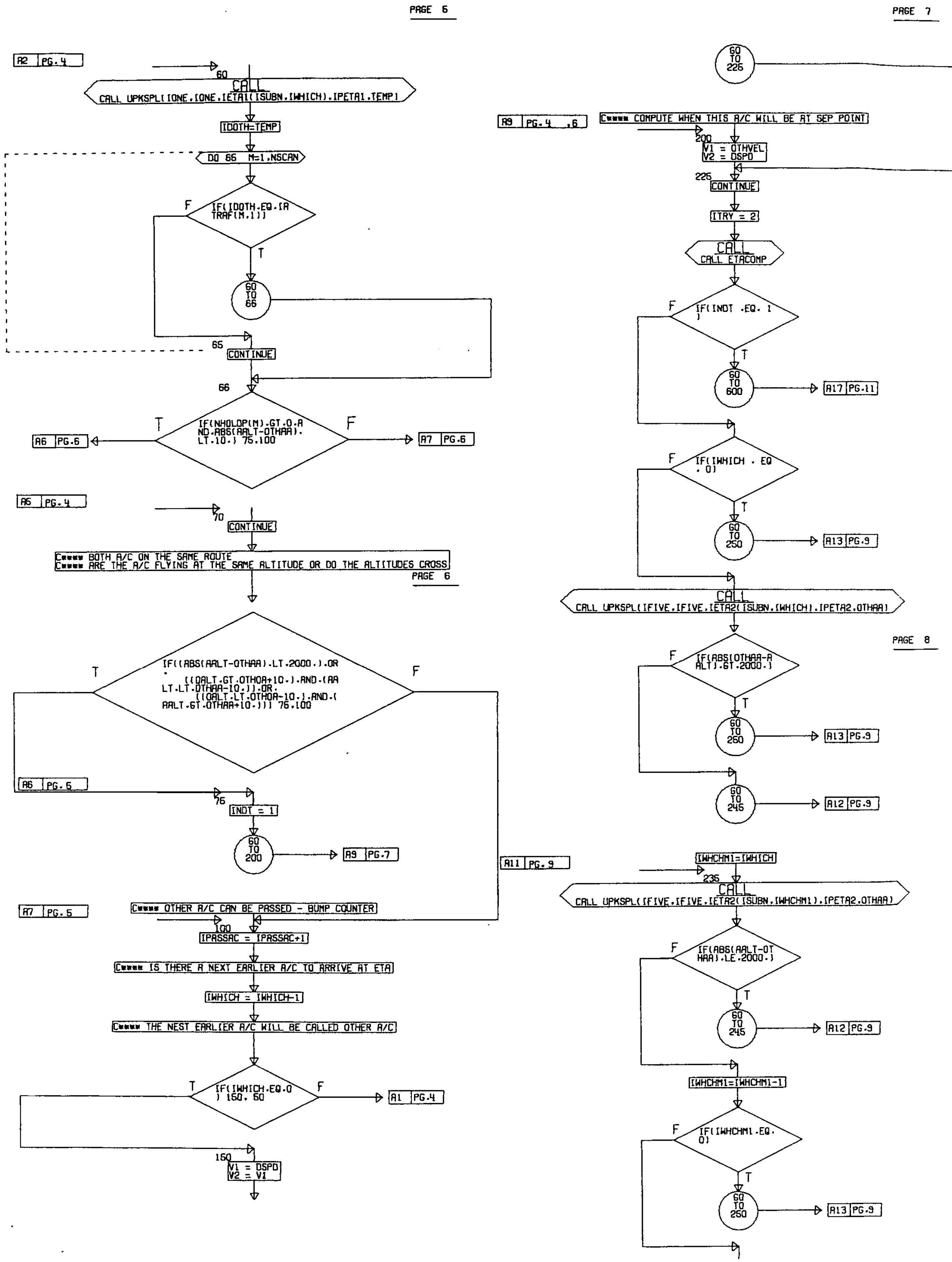
```
*****
* * * REAL-TIME MAIN COMMON IN OVERLAY(4.0) AND ALL SECONDARY OVERLAYS A
* * *     PRIMARY SUBROUTINES OF (4.0)
*****

COMMON/RTMAIN1/
   ASKSPD(35.11).ASPACE1(4.9).ASPACE2(4.9).ASPACE3(4.9).
     CETA.             CONFLCT.       DELH(35).
     HDBEG(35).        HDOWN.         IA(2).
     IAC.              IACTN1(35).    IACTN2(35).
  IACTN3(35). IASKED(35.11). IATAS(35.2).ICNGO.
     IATRAF(35.26).    ICA.           ICACT.
     ICLASS.           ICMSG1(8.5).   ICMSG2(8.5).
     ICMSG3(8.5).      ICOMM(10).     ID(2).
     IDELAY.           IENROT(2.20).  IETAS(35.2).
     IETA1(36.10).     IETA2(36.10).  IE360(60).
     IFCMSG1(8.25).    IFCMSG2(8.25). IFCMSG3(8.25).
     IHOWN.            IOBETA.        IOEETA.
     IOPATH.           IPASSAC.       ISEQ.
     ISEQUE(3.16).     ISWH.          ITOQUE(2.20).
     ITRY.             IWHICH.        KNTP.
     L.                LEMG(5).       NASPACE(4).
     NCMSG(10).        NEMG.          NETA(36).
     NFMSG(10).        NHOLDP(35).    NNA(35).
     NSEQ.             OLDR(35)

COMMON/RTMAIN2/
     OTHAA.            OTHETA.        OTHOA.
     OTHSETA.          RESTOR(35).    ROLL.
     T(35).            TETAE.         TETAL.
     TETAO.            TIME.          V1.
     V2.               XOWN.          XPL(35).
     YOWN.             YPL(35).       ZOWN
   . R9LX(2).          R9LY(2).       R9RX(2).       R9RY(2).
     R15X(2).          R15Y(2).       R21X(2).       R21Y(2).
     XNREF(4).         YNREF(4)

*****

DIMENSION
     IOPTION(9).       OPTION(9).     TEMPA(10)

EQUIVALENCE  (INTEG(41). NSCAN)
```

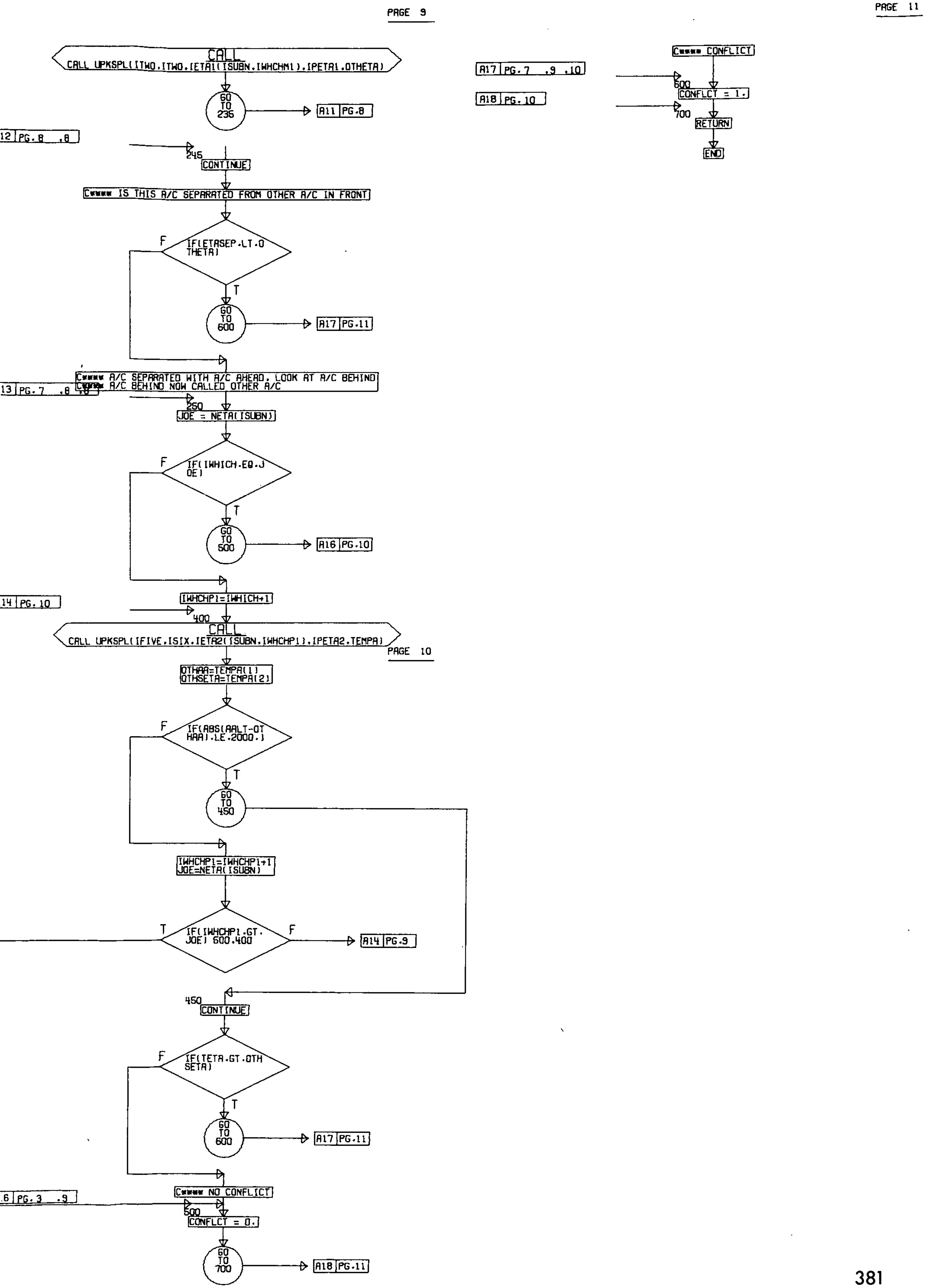


```
EQUIVALENCE
     (RESTOR(01). TETA).              (RESTOR(02). NUMPATH).
     (RESTOR(03). OALT).              (RESTOR(04). AALT).
     (RESTOR(05). ETASEP).            (RESTOR(06). FVEL).
     (RESTOR(07). OSPD).              (RESTOR(08). TX).
     (RESTOR(09). TOTD).              (RESTOR(10). NSEGM).
     (RESTOR(11). DST1).              (RESTOR(12). DST2).
     (RESTOR(13). DST3).              (RESTOR(14). ISUBCK).
     (RESTOR(15). ISUBOP).            (RESTOR(16). ICACT2).
     (RESTOR(17). STEMP).             (RESTOR(18). ISUBN).
     (RESTOR(19). ISUBO).             (RESTOR(20). SEP).
     (RESTOR(21). XLA).               (RESTOR(22). XLB).
     (RESTOR(23). IASPCON).
     (RESTOR(24). IOALT. IOPTION(1). OPTION(1)).
     (RESTOR(25). IDSPD).             (RESTOR(26). IVECT).
     (RESTOR(27). IVECOP).            (RESTOR(28). IDVERG2).
     (RESTOR(29). IALTOP).            (RESTOR(30). ISPDOP).
     (RESTOR(31). ISHOP).             (RESTOR(32). IEHOP)
   . (RESTOR(33). OSPD)
   . (RESTOR(34). TX3).               (RESTOR(35). INDSPD)

IPASSAC = 0

C**** LET A/C NEEDING CLEARANCE BE CALLED THIS A/C. TETA=TIME OF ARRIVAL
C**** LOCATE HOW MANY A/C ARE IN ETA ARRAY AT WHICH CONFLICT CHECK
C****    IS TO BE DONE - SUBSCRIPT OF THIS ETA ARRAY IS ISUBN

IWHICH = NETA(ISUBN)
INDT = 0

C**** ARE THERE ANY A/C IN THIS ETA ARRAY

IF(IWHICH.NE.0
)
   F:
   T: GO TO 50  ->  A1 PG.4

ITRY=2
V1=DSPD
V2=V1

CALL
CALL ETACOMP

GO TO 500  ->  A16 PG.10
```



```
C**** LET THE LAST A/C IN THE ARRAY BE CALLED OTHER A/C
C**** LOCATE ETA OF OTHER A/C

A1 PG.3 .6  ->  50

CALL
CALL UPKSPL(ITWO.ITHREE.IETA1(ISUBN.IWHICH).IPETA1.TEMPA)

C**** WILL THIS A/C ARRIVE AT CHECK POINT BEFORE OTHER

OTHETA = TEMPA(1)
OTHVEL=TEMPA(2)

IF(TETA.GT.OTH
ETA)
   F:
   T: GO TO 200  ->  A9 PG.7

C**** THIS A/C WILL ARRIVE BEFORE OTHER
C**** UNPACK PATH INFO FOR OTHER A/C

CALL
CALL UPKSPL(IONE.ISIX.IETA2(ISUBN.IWHICH).IPETA2.TEMPA)

IOBETA = TEMPA(1)
IOEETA = TEMPA(2)
IOPATH = TEMPA(3)
OTHOA = TEMPA(4)
OTHAA = TEMPA(5)
OTHSETA = TEMPA(6)

C**** IS THIS A/C ON THE SAME ROUTE AS OTHER A/C

IF(ISUBO.NE.IOBETA.
OR.ISUBN.NE.IOEETA.
OR.NUMPATH.NE.IOPAT
H)          60.70
   T:  ->  A2 PG.5
   F:  ->  A5 PG.5
```


PAGE 6
PAGE 7
A2 PG.4
60
CALL
CALL UPKSPL(IONE,IONE,IETA1(ISUBN,IWHICH),IPETA1,TEMP)
IDOTH=TEMP
DO 65 M=1,NSCAN
IF(IDOTH.EQ.IA TRAF(M,1))
F
T
GO TO 66
65
CONTINUE
66
IF(NHOLDP(M).GT.0.A ND.ABS(AALT-OTHAA). LT.10.) 75,100
T
F
A6 PG.6
A7 PG.6
A5 PG.4
70
CONTINUE
C**** BOTH A/C ON THE SAME ROUTE
C**** ARE THE A/C FLYING AT THE SAME ALTITUDE OR DO THE ALTITUDES CROSS
PAGE 6
IF((ABS(AALT-OTHAA).LT.2000.).OR . ((OALT.GT.OTHOA+10.).AND.(AA LT.LT.OTHAA-10.)).OR. ((OALT.LT.OTHOA-10.).AND.(AALT.GT.OTHAA+10.))) 75,100
T
F
A6 PG.5
76
INDT = 1
GO TO 200
A9 PG.7
A7 PG.5
C**** OTHER A/C CAN BE PASSED - BUMP COUNTER
100
IPASSAC = IPASSAC+1
C**** IS THERE A NEXT EARLIER A/C TO ARRIVE AT ETA
IWHICH = IWHICH-1
C**** THE NEST EARLIER A/C WILL BE CALLED OTHER A/C
IF(IWHICH.EQ.0) 150, 50
T
F
A1 PG.4
150
V1 = DSPD
V2 = V1
GO TO 225
A9 PG.4 ,6
C**** COMPUTE WHEN THIS A/C WILL BE AT SEP POINT
200
V1 = OTHVEL
V2 = DSPD
225
CONTINUE
ITRY = 2
CALL
CALL ETACOMP
IF(INDT .EQ. 1)
F
T
GO TO 600
A17 PG.11
IF(IWHICH . EQ . 0)
F
T
GO TO 250
A13 PG.9
CALL
CALL UPKSPL(IFIVE,IFIVE,IETA2(ISUBN,IWHICH),IPETA2,OTHAA)
PAGE 8
IF(ABS(OTHAA-A ALT).GT.2000.)
F
T
GO TO 250
A13 PG.9
GO TO 245
A12 PG.9
A11 PG.9
IWHCHM1=IWHICH
235
CALL
CALL UPKSPL(IFIVE,IFIVE,IETA2(ISUBN,IWHCHM1),IPETA2,OTHAA)
IF(ABS(AALT-OT HAA).LE.2000.)
F
T
GO TO 245
A12 PG.9
IWHCHM1=IWHCHM1-1
IF(IWHCHM1.EQ. 0)
F
T
GO TO 250
A13 PG.9


PAGE 9
CALL
CALL UPKSPL(ITWO,ITWO,IETA1(ISUBN,IWHCHM1),IPETA1,OTHETA)
GO TO 236
A11 PG.8
12 PG.8 .8
245
CONTINUE
C***** IS THIS A/C SEPARATED FROM OTHER A/C IN FRONT
IF(ETASEP.LT.OTHETA)
F
T
GO TO 600
A17 PG.11
C***** A/C SEPARATED WITH A/C AHEAD, LOOK AT A/C BEHIND
C***** A/C BEHIND NOW CALLED OTHER A/C
13 PG.7 .8 .8
250
JOE = NETA(ISUBN)
IF(IWHICH.EQ.JOE)
F
T
GO TO 500
A16 PG.10
IWHCHP1=IWHICH+1
14 PG.10
400
CALL
CALL UPKSPL(IFIVE,ISIX,IETA2(ISUBN,IWHCHP1),IPETA2,TEMPA)
PAGE 10
OTHAA=TEMPA(1)
OTHSETA=TEMPA(2)
IF(ABS(AALT-OTHAA).LE.2000.)
F
T
GO TO 450
IWHCHP1=IWHCHP1+1
JOE=NETA(ISUBN)
IF(IWHCHP1.GT.JOE) 600,400
T
F
A14 PG.9
450
CONTINUE
IF(TETA.GT.OTHSETA)
F
T
GO TO 600
A17 PG.11
C***** NO CONFLICT
16 PG.3 .9
500
CONFLCT = 0.
GO TO 700
A18 PG.11
PAGE 11
A17 PG.7 .9 .10
A18 PG.10
C***** CONFLICT
600
CONFLCT = 1.
700
RETURN
END

PAGE 1

```
SUBROUTINE SPDCNTR

LOGICAL LDISI, LDISO

COMMON / REALTIM/ ADC(32),DAC(64),LDISI(108),LDISO(196),
NOPER,NHOLD,NRESET,NTERM,NPRINT,NREAD

COMMON/TABLES/
   INTEG(80),      TABLE(075),      LOGIC(20)

LOGICAL LOGIC

*****
* * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN
*****

COMMON/MAJOR1/
     ACLS(50),        ALTOPT(3.7.2).  ALTSKD(9).
     ARRMNA.          ARRMXA.         BOUND.
     CGLIDE.          CILS.           CRA20.
     CRA30.           CRB20.          CRB30.
     DGRTD.           DTR.            EHOLDD.
     ENROTA(2).       FTNM.           G.
     GLIDE(2).        HOLDSEP.        IARRN.
     IARRS.           IART(11).       IBF(110).
     ICHKPT(40).      ICNTRA1(85).    ICNTRA2(85).
     IDELT.           IDEPN.          IDEPS.
     IDROUTE(10).     IHPATHE.        IHPATHS.
     INAV(10.5).      IOPTN(55).      IPAPATH(4.5).
     IPASKED(4.7).    IPCHKPT(4.6).   IPCNTR1(4.6).
     IPCNTR2(4.2).    IPORT(4.5).     IPERFR(3.7).
     IPERFS(4.8).     IPERFT(3).      IPETA1(4.4).
     IPETA2(4.6).     IPFTR(3).       IPMSG2(4.6).
     IPMSG3(4.3).     IPNPATH(4.7).   IPNSITE(4.4).
     IPOPL(2).        IPOPPA.         IPOPR(2).
     IPOPTN(4.9).     IPSKED(4.8).    IPTAS(4.5)

COMMON/MAJOR2/
     IPTRAF1(4.4).    IPTRAF2(4.9).   ISDST(17).
     ISKDPU.          ISKED(85).      ISTARTA(21).
     ITANG(2.2).      ITRAF1(102).    ITRAF2(102).
     MAXK.            MF.
     NALTSKD.         NCP.            NDEPR.
     NNAV(5).         NNPATH(50).     NOUT(20).
     NSPDW.           OUT(26).        PI.
     PPMX(3).         PPMY(3).        RCP(2).
     RTD.             RWYX(4.2).      RWYY(4.2).
     RWYZ(2).         SHOLDD.         SPDSTD(9.2).
     SQX(6).          SQY(6).         TRIX(5).
     TRIY(5).         XA(2).          XC(2).
     XDILS(2).        XDIST(2).       XFINAL(2).
     XRPOS(2).        YA(2).          YC(2).
     YRPOS(2).        RAD(3).         SIG(3).        IARG
   . IETCOOR(36).     IPETC(4.2).     IACFPER(3)

*****
```

PAGE 2

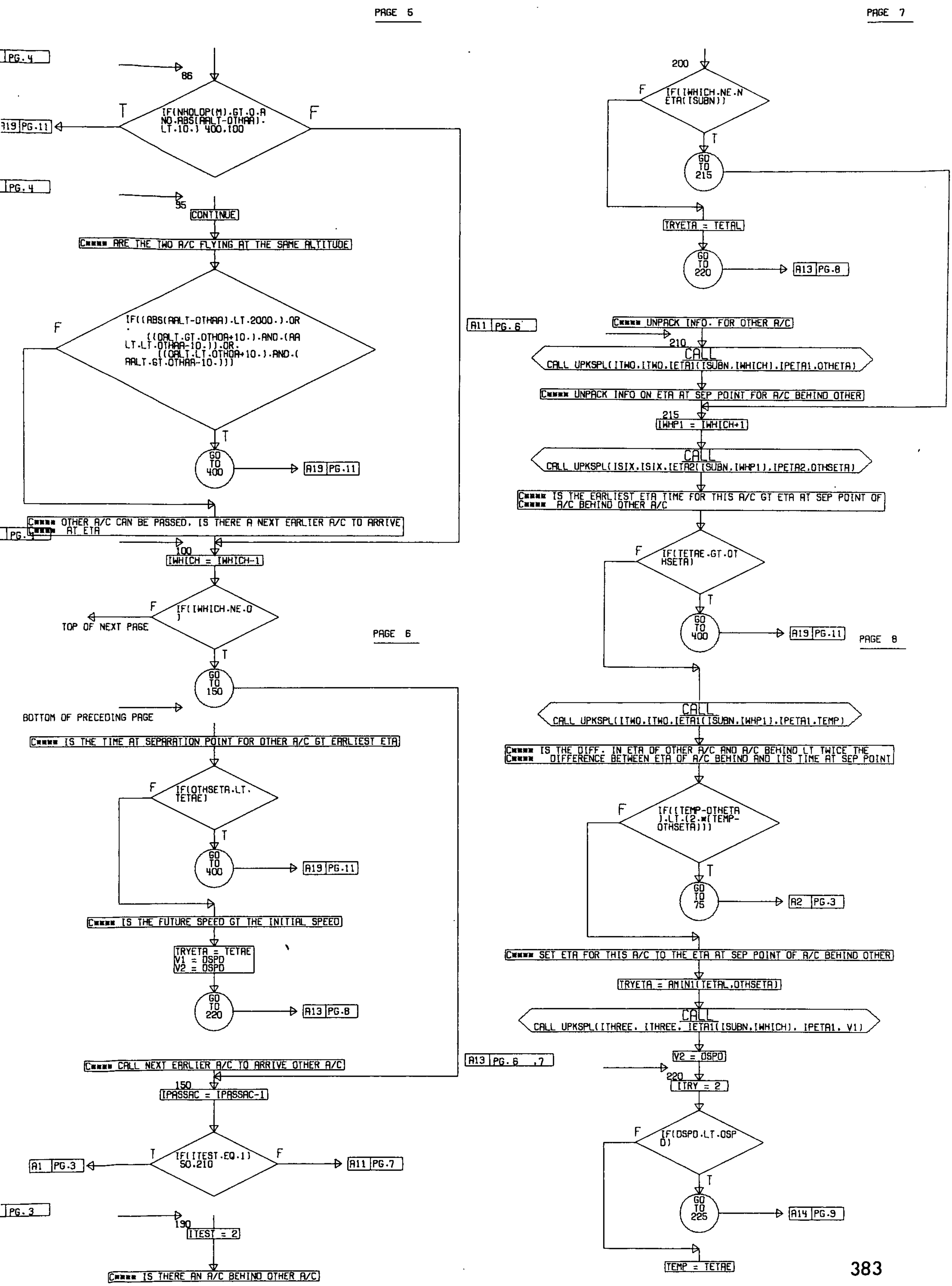
```
COMMON/TOES/ IONE.ITWO.ITHREE.IFOUR.IFIVE.ISIX.ISEVEN.IEIGHT.ININE
.ITEN

*****
* * * REAL-TIME MAIN COMMON IN OVERLAY(4.0) AND ALL SECONDARY OVERLAYS A
* * *     PRIMARY SUBROUTINES OF (4.0)
*****

COMMON/RTMAIN1/
   ASKSPD(35.11).ASPACE1(4.9).ASPACE2(4.9).ASPACE3(4.9).
    CETA.            CONFLCT.        DELH(35).
    HOBEG(35).       HDOWN.          IA(2).
    IAC.             IACTN1(35).     IACTN2(35).
  IACTN3(35).  IASKED(35.11).  IATAS(35.2).ICNGO.
    IATRAF(35.26).  ICA.             ICACT.
    ICLASS.          ICMSG1(8.5).    ICMSG2(8.5).
    ICMSG3(8.5).     ICOMM(10).      ID(2).
    IDELAY.          IENROT(2.20).   IETAS(35.2).
    IETA1(36.10).    IETA2(36.10).   IE360(60).
    IFCMSG1(8.25).   IFCMSG2(8.25).  IFCMSG3(8.25).
    IHOWN.           IOBETA.         IOEETA.
    IOPATH.          IPASSAC.        ISEQ.
    ISEQUE(3.16).    ISW4.           ITOQUE(2.20).
    ITRY.            IWHICH.         KNTP.
    L.               LEMG(5).        NASPACE(4).
    NCMSG(10).       NEMG.           NETA(36).
    NFMSG(10).       NHOLDP(35).     NNA(35).
    NSEQ.            OLOR(35)

COMMON/RTMAIN2/
    OTHAA.           OTHETA.         OTHOA.
    OTHSETA.         RESTOR(35).     ROLL.
    T(35).           TETAE.          TETAL.
    TETAO.           TIME.           V1.
    V2.              XOWN.           XPL(35).
    YOWN.            YPL(35).        ZOWN
  . R9LX(2).         R9LY(2).        R9RX(2).       R9RY(2).
    R15X(2).         R15Y(2).        R21X(2).       R21Y(2).
    XNREF(4).        YNREF(4)

*****

EQUIVALENCE  (INTEG(41). NSCAN)

DIMENSION
   IOPTION(9).    OPTION(9).    TEMPA(10)
```

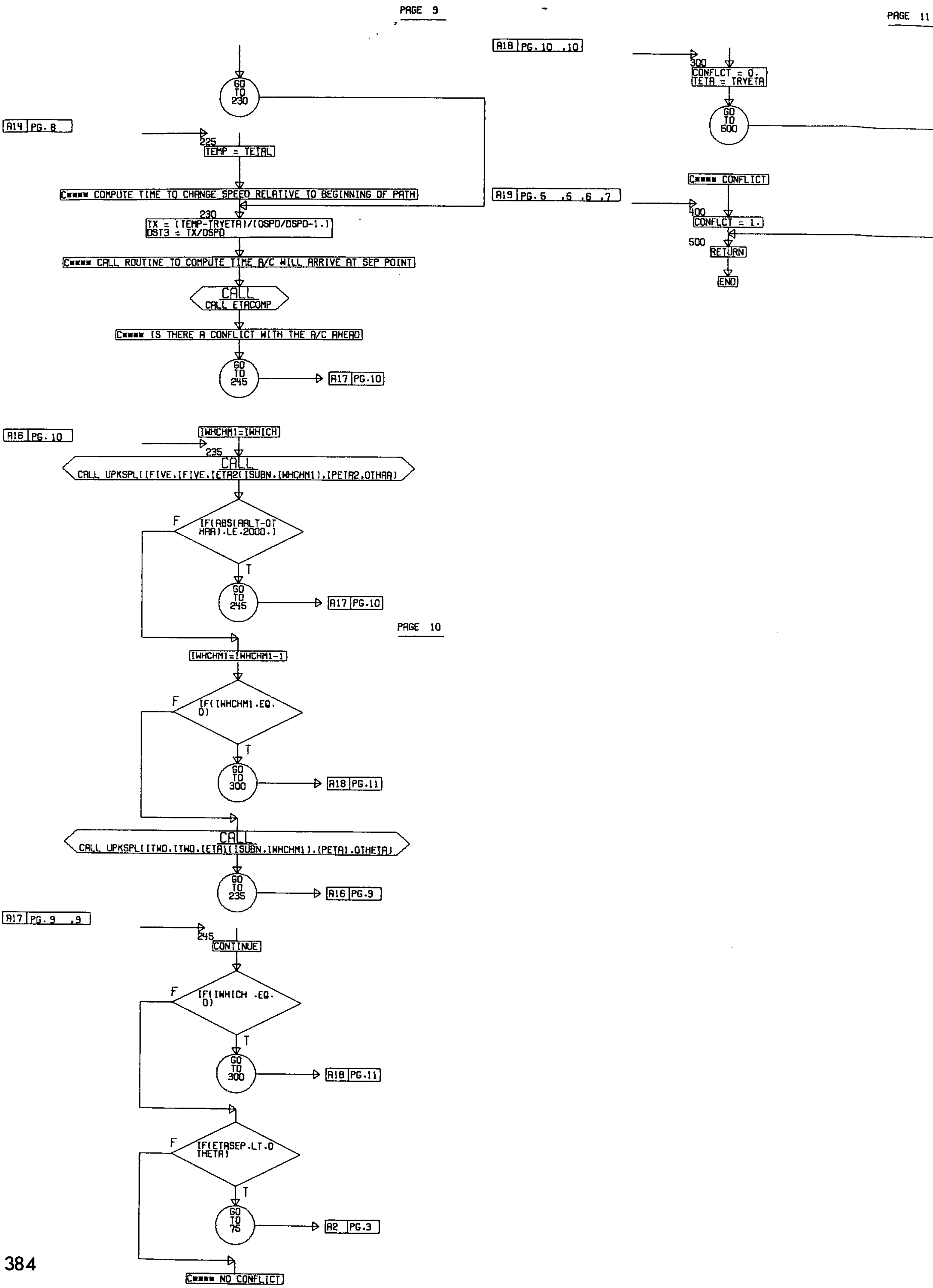

PAGE 3

```
EQUIVALENCE
    (RESTOR(01). TETA).             (RESTOR(02). NUMPATH).
    (RESTOR(03). OALT).             (RESTOR(04). AALT).
    (RESTOR(05). ETASEP).           (RESTOR(06). FVEL).
    (RESTOR(07). DSPD).             (RESTOR(08). TX).
    (RESTOR(09). TOTO).             (RESTOR(10). NSEGM).
    (RESTOR(11). DST1).             (RESTOR(12). DST2).
    (RESTOR(13). DST3).             (RESTOR(14). ISUBCK).
    (RESTOR(15). ISUBOP).           (RESTOR(16). ICACT2).
    (RESTOR(17). STEMP).            (RESTOR(18). ISUBN).
    (RESTOR(19). ISUBO).            (RESTOR(20). SEP).
    (RESTOR(21). XLA).              (RESTOR(22). XLB).
    (RESTOR(23). IASPCON).
    (RESTOR(24). IOALT. IOPTION(1). OPTION(1)).
    (RESTOR(25). IOSPD).            (RESTOR(26). IVECT).
    (RESTOR(27). IVECOP).           (RESTOR(28). IOVERG2).
    (RESTOR(29). IALTOP).           (RESTOR(30). ISPDOP).
    (RESTOR(31). ISHOP).            (RESTOR(32). IEHOP)
  . (RESTOR(33). OSPD)
  . (RESTOR(34). TX3).              (RESTOR(35). INDSPD)

C**** A/C TRYING TO FIND SLOT TO MERGE INTO CALLED THIS A/C
C**** LET LAST A/C IN ETA ARRAY BE CALLED OTHER A/C

ITEST = 1
IWHICH = NETA(ISUBN)

A1 PG.6
 50
CALL
CALL UPKSPL(ITWO.ITWO.IETA1(ISUBN.IWHICH).IPETA1.OTHETA)

C**** IS THE LATEST POSSIBLE ETA FOR THIS A/C GE ETA OF OTHER A/C AT
C****   CHECK POINT

IF(TETAL.GE.OTHETA)      F
   T
GO TO 190  -->  A9 PG.6

A2 PG.6 ,10
C**** THIS A/C CAN ARRIVE EARLIER THAN OTHER. CAN OTHER A/C BE PASSED
 75

IF(IPASSAC.GT.0)     F --> TOP OF NEXT PAGE
   T
GO TO 100  -->  A7 PG.5
```

PAGE 4

```
BOTTOM OF PRECEDING PAGE -->

CALL
CALL UPKSPL(IONE.ISIX.IETA2(ISUBN.IWHICH).IPETA2.TEMPA)

IOBETA = TEMPA(1)
IOEETA = TEMPA(2)
IOPATH = TEMPA(3)
OTHOA = TEMPA(4)
OTHAA = TEMPA(5)
OTHSETA = TEMPA(6)

C**** IS THIS A/C ON THE SAME ROUTE AS OTHER

IF(ISUBO.NE.IOBETA.OR.ISUBN.NE.IOEETA.OR.NUMPATH.NE.IOPATH) 80.95
   T                                            F --> A6 PG.5
 80
CALL
CALL UPKSPL(IONE.IONE.IETA1(ISUBN.IWHICH).IPETA1.TEMP)

IOOTH=TEMP

DO 85  M=1.NSCAN

IF(IOOTH.EQ.IATRAF(M.1))     F
   T
GO TO 86  -->  A5 PG.5

 85
CONTINUE
```


PAGE 5
PG.4
86
IF(NHOLOP(M).GT.0.AND.ABS(AALT-OTHAA).LT.10.) 400,100
T
F
A19 PG.11
PG.4
95
CONTINUE
C**** ARE THE TWO A/C FLYING AT THE SAME ALTITUDE
IF((ABS(AALT-OTHAA).LT.2000.).OR.((OALT.GT.OTHOA+10.).AND.(AALT.LT.OTHAA-10.)).OR.((OALT.LT.OTHOA+10.).AND.(AALT.GT.OTHAA-10.)))
F
T
GO TO 400
A19 PG.11
C**** OTHER A/C CAN BE PASSED, IS THERE A NEXT EARLIER A/C TO ARRIVE
C**** AT ETA
PG.4
100
IWHICH = IWHICH-1
IF(IWHICH.NE.0)
F
TOP OF NEXT PAGE
T
GO TO 150
PAGE 6
BOTTOM OF PRECEDING PAGE
C**** IS THE TIME AT SEPARATION POINT FOR OTHER A/C GT EARLIEST ETA
IF(OTHSETA.LT.TETAE)
F
T
GO TO 400
A19 PG.11
C**** IS THE FUTURE SPEED GT THE INITIAL SPEED
TRYETA = TETAE
V1 = OSPD
V2 = OSPD
GO TO 220
A13 PG.8
C**** CALL NEXT EARLIER A/C TO ARRIVE OTHER A/C
150
IPASSAC = IPASSAC-1
IF(ITEST.EQ.1) 50,210
T
F
A1 PG.3
A11 PG.7
PG.3
190
ITEST = 2
C**** IS THERE AN A/C BEHIND OTHER A/C
PAGE 7
200
IF(IWHICH.NE.NETA(ISUBN))
F
T
GO TO 215
TRYETA = TETAL
GO TO 220
A13 PG.8
A11 PG.6
C**** UNPACK INFO. FOR OTHER A/C
210
CALL
CALL UPKSPL(ITWO,ITWO,IETA1(ISUBN,IWHICH),IPETA1,OTHETA)
C**** UNPACK INFO ON ETA AT SEP POINT FOR A/C BEHIND OTHER
215
IWHP1 = IWHICH+1
CALL
CALL UPKSPL(ISIX,ISIX,IETA2(ISUBN,IWHP1),IPETA2,OTHSETA)
C**** IS THE EARLIEST ETA TIME FOR THIS A/C GT ETA AT SEP POINT OF
C**** A/C BEHIND OTHER A/C
IF(TETAE.GT.OTHSETA)
F
T
GO TO 400
A19 PG.11
PAGE 8
CALL
CALL UPKSPL(ITWO,ITWO,IETA1(ISUBN,IWHP1),IPETA1,TEMP)
C**** IS THE DIFF. IN ETA OF OTHER A/C AND A/C BEHIND LT TWICE THE
C**** DIFFERENCE BETWEEN ETA OF A/C BEHIND AND ITS TIME AT SEP POINT
IF((TEMP-OTHETA).LT.(2.*(TEMP-OTHSETA)))
F
T
GO TO 75
A2 PG.3
C**** SET ETA FOR THIS A/C TO THE ETA AT SEP POINT OF A/C BEHIND OTHER
TRYETA = AMIN1(TETAL,OTHSETA)
CALL
CALL UPKSPL(ITHREE, ITHREE, IETA1(ISUBN,IWHICH), IPETA1, V1)
V2 = OSPD
A13 PG.6 ,7
220
ITRY = 2
IF(OSPD.LT.OSPD)
F
T
GO TO 225
A14 PG.9
TEMP = TETAE


PAGE 9
A14 PG. 8
GO TO 230
225
TEMP = TETAL
C***** COMPUTE TIME TO CHANGE SPEED RELATIVE TO BEGINNING OF PATH
230
TX = (TEMP-TRYETA)/(DSPD/DSPD-1.)
DST3 = TX/DSPD
C***** CALL ROUTINE TO COMPUTE TIME A/C WILL ARRIVE AT SEP POINT
CALL
CALL ETACOMP
C***** IS THERE A CONFLICT WITH THE A/C AHEAD
GO TO 245
A17 PG.10
A16 PG. 10
IWHCHM1=IWHICH
235
CALL
CALL UPKSPL(IFIVE.IFIVE.IETA2(ISUBN.IWHCHM1).IPETA2.OTHAA)
IF(ABS(AALT-OTHAA).LE.2000.)
F
T
GO TO 245
A17 PG.10
PAGE 10
IWHCHM1=IWHCHM1-1
IF(IWHCHM1.EQ.0)
F
T
GO TO 300
A18 PG.11
CALL
CALL UPKSPL(ITWO.ITWO.IETA1(ISUBN.IWHCHM1).IPETA1.OTHETA)
GO TO 235
A16 PG.9
A17 PG. 9 ,9
245
CONTINUE
IF(IWHICH .EQ. 0)
F
T
GO TO 300
A18 PG.11
IF(ETASEP.LT.OTHETA)
F
T
GO TO 75
A2 PG.3
C***** NO CONFLICT
A18 PG. 10 ,10
A19 PG. 5 ,5 ,6 ,7
PAGE 11
300
CONFLCT = 0.
TETA = TRYETA
GO TO 500
C***** CONFLICT
400
CONFLCT = 1.
500
RETURN
END

PAGE 1

```
SUBROUTINE COMETA(ETAPT1,ETAPT2)

LOGICAL LDISI, LDISO

COMMON / REALTIM/ ADC(32),DAC(64),LDISI(108),LDISO(196),
NOPER,NHOLD,NRESET,NTERM,NPRINT,NREAD

COMMON/TABLES/
    INTEG(80),      TABLE(075),     LOGIC(20)

LOGICAL LOGIC

*****
* * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN
*****

COMMON/MAJOR1/
    ACLS(50),       ALTOPT(3,7,2),  ALTSKD(9),
    ARRMNA,         ARRMXA,         BOUND,
    CGLIDE,         CILS,           CRA20,
    CRA30,          CRB20,          CRB30,
    DGRTD,          DIR,            EHOLDD,
    ENROTA(2),      FTNM,           G,
    GLIDE(2),       HOLDSEP,        IARRN,
    IARRS,          IART(11),       IBF(110),
    ICHKPT(40),     ICNTRA1(85),    ICNTRA2(85),
    IDELT,          IDEPN,          IDEPS,
    IDROUTE(10),    IHPATHE,        IHPATHS,
    INAV(10,5),     IOPTN(55),      IPAPATH(4,5),
    IPASKED(4,7),   IPCHKPT(4,6),   IPCNTR1(4,6),
    IPCNTR2(4,2),   IPDRT(4,5),     IPERFR(3,7),
    IPERFS(4,8),    IPERFT(3),      IPETA1(4,4),
    IPETA2(4,6),    IPFTR(3),       IPMSG2(4,6),
    IPMSG3(4,3),    IPNPATH(4,7),   IPNSITE(4,4),
    IPOPL(2),       IPOPPA,         IPOPR(2),
    IPOPTN(4,9),    IPSKED(4,8),    IPTAS(4,5)

COMMON/MAJOR2/
    IPTRAF1(4,4),   IPTRAF2(4,9),   ISDST(17),
    ISKDPU,         ISKED(85),      ISTARTA(21),
    ITANG(2,2),     ITRAF1(102),    ITRAF2(102),
    MAXK,           MF,
    NALTSKD,        NCP,            NDEPR,
    NNAV(5),        NNPATH(50),     NOUT(20),
    NSPDW,          OUT(26),        PI,
    PPMX(3),        PPMY(3),        RCP(2),
    RTD,            RWYX(4,2),      RWYY(4,2),
    RWYZ(2),        SHOLDD,         SPDSTD(9,2),
    SQX(6),         SQY(6),         TRIX(5),
    TRIY(5),        XA(2),          XC(2),
    XDILS(2),       XDIST(2),       XFINAL(2),
    XRPOS(2),       YA(2),          YC(2),
    YRPOS(2),       RAD(3),         SIG(3),         IARG
  . IETCOOR(36),    IPETC(4,2),     IACFPER(3)

*****
```

PAGE 2

```
COMMON/TOES/ IONE,ITWO,ITHREE,IFOUR,IFIVE,ISIX,ISEVEN,IEIGHT,ININE
.ITEN

*****
* * * REAL-TIME MAIN COMMON IN OVERLAY(4,0) AND ALL SECONDARY OVERLAYS A
* * *      PRIMARY SUBROUTINES OF (4,0)
*****

COMMON/RTMAIN1/
  ASKSPD(35,11),ASPACE1(4,9),ASPACE2(4,9),ASPACE3(4,9),
    CETA,           CONFLCT,        DELH(35),
    HDBEG(35),      HDOWN,          IA(2),
    IAC,            IACTN1(35),     IACTN2(35),
 IACTN3(35),  IASKED(35,11),  IATAS(35,2),ICNGO,
    IATRAF(35,26),  ICA,            ICACT,
    ICLASS,         ICMSG1(8,5),    ICMSG2(8,5),
    ICMSG3(8,5),    ICOMM(10),      ID(2),
    IDELAY,         IENROT(2,20),   IETAS(35,2),
    IETA1(36,10),   IETA2(36,10),   IE360(60),
    IFCMSG1(8,25),  IFCMSG2(8,25),  IFCMSG3(8,25),
    IHOWN,          IOBETA,         IOEETA,
    IOPATH,         IPASSAC,        ISEQ,
    ISEQUE(3,16),   ISWH,           ITOQUE(2,20),
    ITRY,           IWHICH,         KNTP,
    L,              LEMG(5),        NASPACE(4),
    NCMSG(10),      NEMG,           NETA(36),
    NFMSG(10),      NHOLDP(35),     NNA(35),
    NSEQ,           OLDR(35)

COMMON/RTMAIN2/
    OTHAA,          OTHETA,         OTHOA,
    OTHSETA,        RESTOR(35),     ROLL,
    T(35),          TETAE,          TETAL,
    TETAO,          TIME,           V1,
    V2,             XOWN,           XPL(35),
    YOWN,           YPL(35),        ZOWN
  . R9LX(2),        R9LY(2),        R9RX(2),        R9RY(2),
    R15X(2),        R15Y(2),        R21X(2),        R21Y(2),
    XNREF(4),       YNREF(4)

*****

DIMENSION
    IOPTION(9),     OPTION(9),      IPERFR6(3),     IPERFR7(3)

EQUIVALENCE
    (IPERFR(1,6), IPERFR6(1)),      (IPERFR(1,7), IPERFR7(1))
```


PAGE 3

```
EQUIVALENCE
    (RESTOR(01), TETA),             (RESTOR(02), NUMPATH),
    (RESTOR(03), OALT),             (RESTOR(04), AALT),
    (RESTOR(05), ETASEP),           (RESTOR(06), FVEL),
    (RESTOR(07), OSPD),             (RESTOR(08), TX),
    (RESTOR(09), TOTD),             (RESTOR(10), NSEGM),
    (RESTOR(11), DST1),             (RESTOR(12), DST2),
    (RESTOR(13), DST3),             (RESTOR(14), ISUBCK),
    (RESTOR(15), ISUBOP),           (RESTOR(16), ICACT2),
    (RESTOR(17), STEMP),            (RESTOR(18), ISUBN),
    (RESTOR(19), ISUBO),            (RESTOR(20), SEP),
    (RESTOR(21), XLA),              (RESTOR(22), XLB),
    (RESTOR(23), IASPCON),
    (RESTOR(24), IDALT, IOPTION(1), OPTION(1)),
    (RESTOR(25), IDSPD),            (RESTOR(26), IVECT),
    (RESTOR(27), IVECOP),           (RESTOR(28), IDVERG2),
    (RESTOR(29), IALTOP),           (RESTOR(30), ISPDOP),
    (RESTOR(31), ISHOP),            (RESTOR(32), IEHOP)
  . (RESTOR(33), OSPD)
  . (RESTOR(34), TX3),              (RESTOR(35), INDSPD)

COMMON/B1/ VI, VF, D, ACCEL, TVI

IF(VI-VF)10,40,25
    -  -> 10
    0  -> 40
    +  -> 25

C**** A/C DESIRES TO ACCELERATE
10  CALL
    CALL UPKUNI(ICLASS,IEIGHT,IPERFR6,1.,ACCEL)
    GO TO 50          -> A4  PG.4

C**** A/C DESIRES TO DECELERATE
25  CALL
    CALL UPKUNI(ICLASS,IEIGHT,IPERFR7,-1.,ACCEL)
    GO TO 50          -> A4  PG.4
```

PAGE 4

```
C**** A/C HOLDS SPEED CONSTANT
40  ETAPT1 = D/VF
    ETAPT2 = ETAPT1
    TVI = VI
    GO TO 200

A4  PG.3  ,3
C**** CHANGE UNITS OF RATE FROM KTS/MIN TO NM/SEC/SEC
50  ACCEL = ACCEL/3600./60.

C**** COMPUTE DISTANCE AND TIME REQUIRED TO CHANGE SPEED
    TA = (VF-VI)/ACCEL
    DA = VI*TA+ACCEL/2.*TA*TA

IF(DA.GT.D)
    T -> GO TO 100
    F ->

C**** SPEED CHANGE WILL BE COMPLETE BEFORE END OF DISTANCE TO TRAVEL
    ETAPT1 = 1./VF*(D-DA)+TA
    ETAPT2 = 1./VI*(D-DA)+TA
    TVI = VF
    GO TO 200

C**** SPEED CHANGE WILL NOT BE COMPLETE BEFORE END OF DISTANCE TO TRAVEL
100 TVI = SQRT(VI*VI+2.*ACCEL*D)
    ETAPT1 = (-VI+TVI)/ACCEL
    ETAPT2 = ETAPT1

200 RETURN

END
```

PAGE 1

```
SUBROUTINE CRTDSPL

LOGICAL LDISI, LDISO

COMMON / REALTIM/ ADC(32).DAC(64).LDISI(108).LDISO(196).
NOPER.NHOLD.NRESET.NTERM.NPRINT.NREAD

COMMON/TABLES/
    INTEG(80).      TABLE(075).     LOGIC(20)

LOGICAL LOGIC

*****
* * * MAJOR COMMON BLOCK CONTAINED IN ALL OVERLAYS AND PRIMARY SUBROUTIN
*****

COMMON/MAJOR1/
     ACLS(50).       ALTOPT(3.7.2). ALTSKD(9).
     ARRMNA.         ARRMXA.        BOUND.
     CGLIDE.         CILS.          CRA20.
     CRA30.          CRB20.         CRB30.
     DGRTD.          DTR.           EHOLDD.
     ENROTA(2).      FTNM.          G.
     GLIDE(2).       HOLDSEP.       IARRN.
     IARRS.          IART(11).      IBF(110).
     ICHKPT(40).     ICNTRA1(85).   ICNTRA2(85).
     IDELT.          IDEPN.         IDEPS.
     IDROUTE(10).    IHPATHE.       IHPATHS.
     INAV(10.5).     IOPTN(55).     IPAPATH(4.5).
     IPASKED(4.7).   IPCHKPT(4.6).  IPCNTR1(4.6).
     IPCNTR2(4.2).   IPDRT(4.5).    IPERFR(3.7).
     IPERFS(4.8).    IPERFT(3).     IPETA1(4.4).
     IPETA2(4.6).    IPFTR(3).      IPMSG2(4.6).
     IPMSG3(4.3).    IPNPATH(4.7).  IPNSITE(4.4).
     IPOPL(2).       IPOPPA.        IPOPR(2).
     IPOPTN(4.9).    IPSKED(4.8).   IPTAS(4.5)

COMMON/MAJOR2/
     IPTRAF1(4.4).   IPTRAF2(4.9).  ISDST(17).
     ISKDPU.         ISKED(85).     ISTARTA(21).
     ITANG(2.2).     ITRAF1(102).   ITRAF2(102).
     MAXK.           MF.
     NALTSKD.        NCP.           NDEPR.
     NNAV(5).        NNPATH(50).    NOUT(20).
     NSPDW.          OUT(26).       PI.
     PPMX(3).        PPMY(3).       RCP(2).
     RTD.            RWYX(4.2).     RWYY(4.2).
     RWYZ(2).        SHOLDD.        SPDSTD(9.2).
     SQX(6).         SQY(6).        TRIX(5).
     TRIY(5).        XA(2).         XC(2).
     XDILS(2).       XDIST(2).      XFINAL(2).
     XRPOS(2).       YA(2).         YC(2).
     YRPOS(2).       RAD(3).        SIG(3).       IARG
   . IETCOOR(36).    IPETC(4.2).    IACFPER(3)

*****
```

PAGE 2

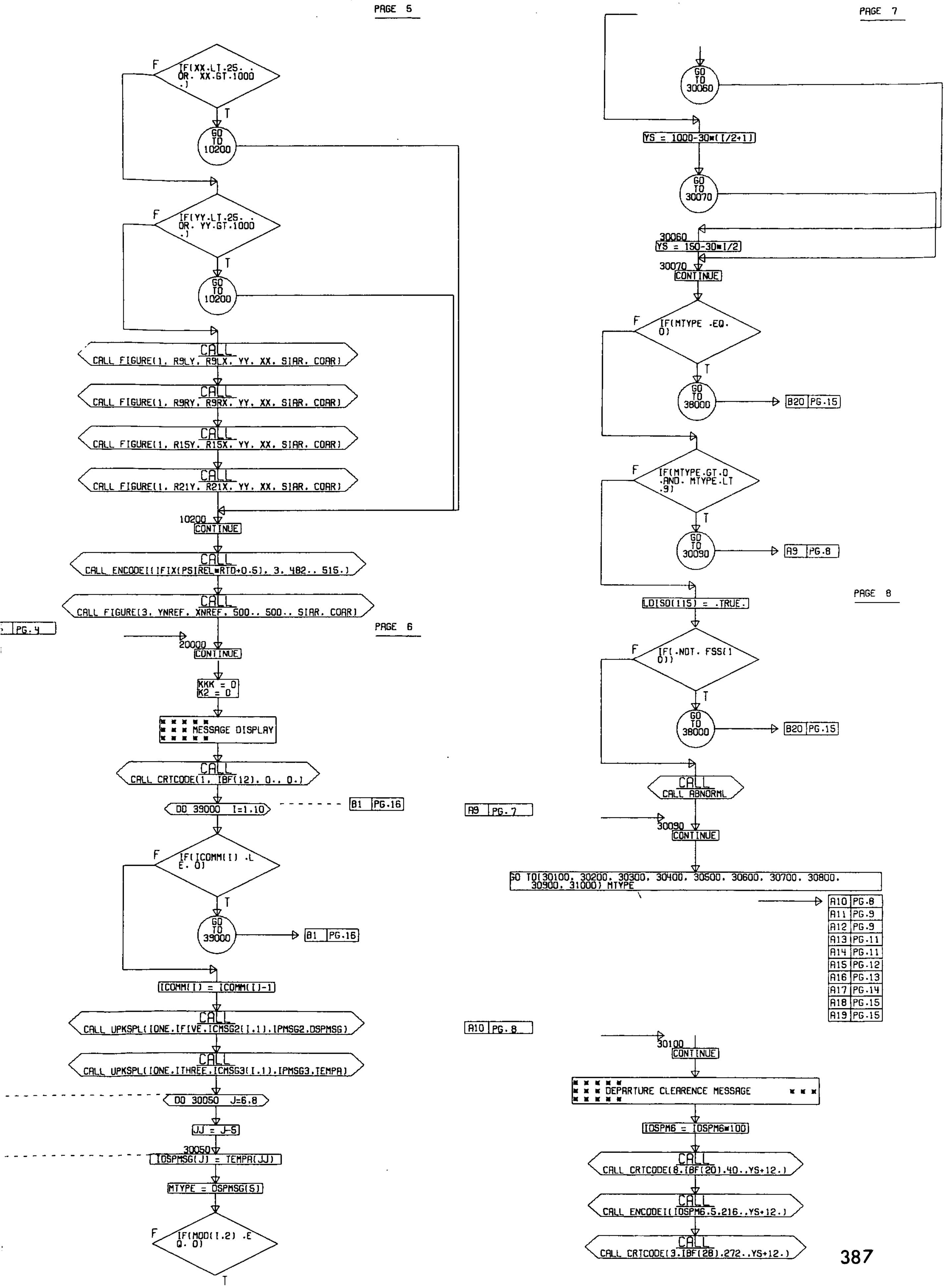
```
COMMON/TOES/ IONE.ITWO.ITHREE.IFOUR.IFIVE.ISIX.ISEVEN.IEIGHT.ININE
.ITEN

*****
* * * REAL-TIME MAIN COMMON IN OVERLAY(4.0) AND ALL SECONDARY OVERLAYS A
* * *      PRIMARY SUBROUTINES OF (4.0)
*****

COMMON/RTMAIN1/
   ASKSPD(35.11).ASPACE1(4.9).ASPACE2(4.9).ASPACE3(4.9).
     CETA.           CONFLCT.       DELH(35).
     HDBEG(35).      HDOWN.         IA(2).
     IAC.            IACTN1(35).    IACTN2(35).
  IACTN3(35). IASKED(35.11). IATAS(35.2).ICNGO.
     IATRAF(35.26).  ICA.           ICACT.
     ICLASS.         ICMSG1(8.5).   ICMSG2(8.5).
     ICMSG3(8.5).    ICOMM(10).     ID(2).
     IDELAY.         IENROT(2.20).  IETAS(35.2).
     IETA1(36.10).   IETA2(36.10).  IE360(60).
     IFCMSG1(8.25).  IFCMSG2(8.25). IFCMSG3(8.25).
     IHOWN.          IOBETA.        IOEETA.
     IOPATH.         IPASSAC.       ISEQ.
     ISEQUE(3.16).   ISW4.          ITOQUE(2.20).
     ITRY.           IWHICH.        KNTP.
     L.              LEMG(5).       NASPACE(4).
     NCMSG(10).      NEMG.          NETA(36).
     NFMSG(10).      NHOLDP(35).    NNA(35).
     NSEQ.           OLDR(35)

COMMON/RTMAIN2/
     OTHAA.          OTHETA.        OTHOA.
     OTHSETA.        RESTOR(35).    ROLL.
     T(35).          TETAE.         TETAL.
     TETAO.          TIME.          V1.
     V2.             XOWN.          XPL(35).
     YOWN.           YPL(35).       ZOWN
   . R9LX(2).        R9LY(2).       R9RX(2).      R9RY(2).
     R15X(2).        R15Y(2).       R21X(2).      R21Y(2).
     XNREF(4).       YNREF(4)

COMMON/XFRM/
     XX.        YY.        SIAR.      COAR

*****
* * * TABLE. INTEG. AND LOGIC EQIV. FOR CRTDSPL

EQUIVALENCE  (INTEG(13). IOPT).
    (TABLE(07). XREL).                (TABLE(08). YREL).
    (TABLE(09). ZREL).                (TABLE(10). PSIREL)

EQUIVALENCE
     (INTEG(08).        INC)      .(INTEG(09).         INCC)
    .(INTEG(27).        KKK)      .(INTEG(28).           K1)
    .(INTEG(29).         K2)      .(INTEG(34).      NCONTRL)
    .(INTEG(40).         IT)      .(INTEG(41).        NSCAN)
    .(INTEG(45).  ISTK)
  . (TABLE(13).  CRTSCAL)
  . (TABLE(31).  DELX1).              (TABLE(32). DELX2)
  . (TABLE(33).  DELX3).              (TABLE(34). DELY1)
  . (TABLE(35).  DELY2)
  . (TABLE(49).  XBOX).               (TABLE(50). YBOX)

*****
* * * DIMENSIONS AND EQUIVALENCES FOR CRTDSPL
*****
```

PAGE 3

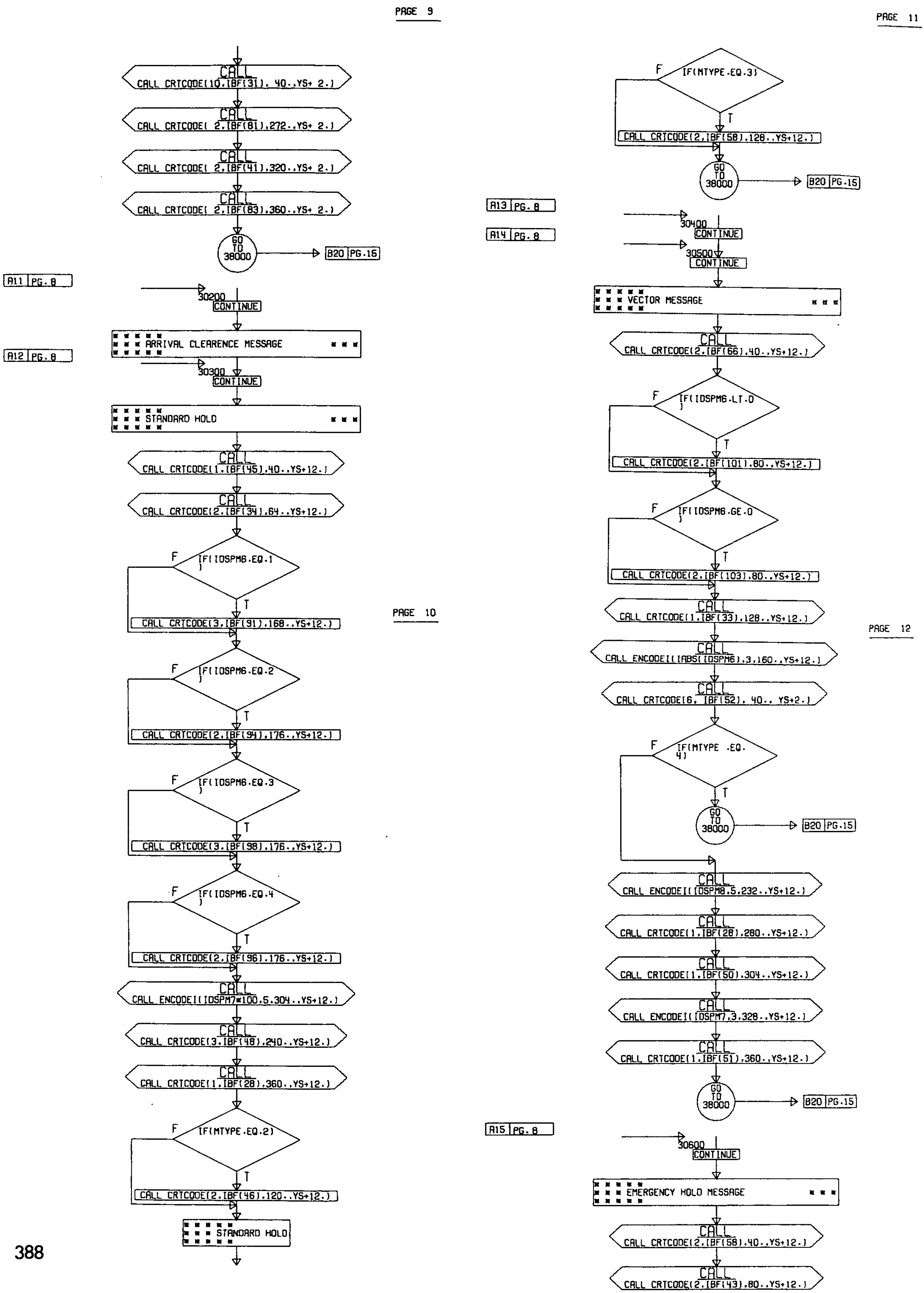
```
DIMENSION
     ATRAF(35.26).   IATRAF1(35).   ATRAF2(35).    ATRAF3(35).
     ATRAF4(35).     ATRAF5(35).    ATRAF6(35).    IATRAF7(35).
     IATRF16(35).
     FSS(16).        DSPMSG(8).     IDSPMSG(8)

DIMENSION TEMPA(3)
LOGICAL FSS

EQUIVALENCE
     (ATRAF(1.1).    IATRAF(1.1)).  (IATRAF(1.1).  IATRAF1(1)).
     (ATRAF(1.2).    ATRAF2(1)).    (ATRAF(1.3).   ATRAF3(1)).
     (ATRAF(1.4).    ATRAF4(1)).    (ATRAF(1.5).   ATRAF5(1)).
     (ATRAF(1.6).    ATRAF6(1)).    (IATRAF(1.7).  IATRAF7(1)).
     (IATRAF(1.16).  IATRF16(1)).
     (LDISI(33).     FSS(1)).       (IDSPMSG(1).   DSPMSG(1)).
     (IDSPMSG(4).    IDSPM4).       (IDSPMSG(6).   IDSPM6).
     (IDSPMSG(7).    IDSPM7).       (IDSPMSG(8).   IDSPM8)

*****
* * * * * * * * BEGIN CRT DISPLAY HERE * * * * * * * * * * * * * * * * *

IF(KKK) 90000. 50000. 10000
   -  -> B12 PG.22
   0  -> B2 PG.16
   +

1000
CALL
CALL SEND

10000
CALL
CALL ENABLE(1000S)

IF(ISTK .EQ. 0
)
   F
   T  -> GO TO 10100  -> A3 PG.4
```

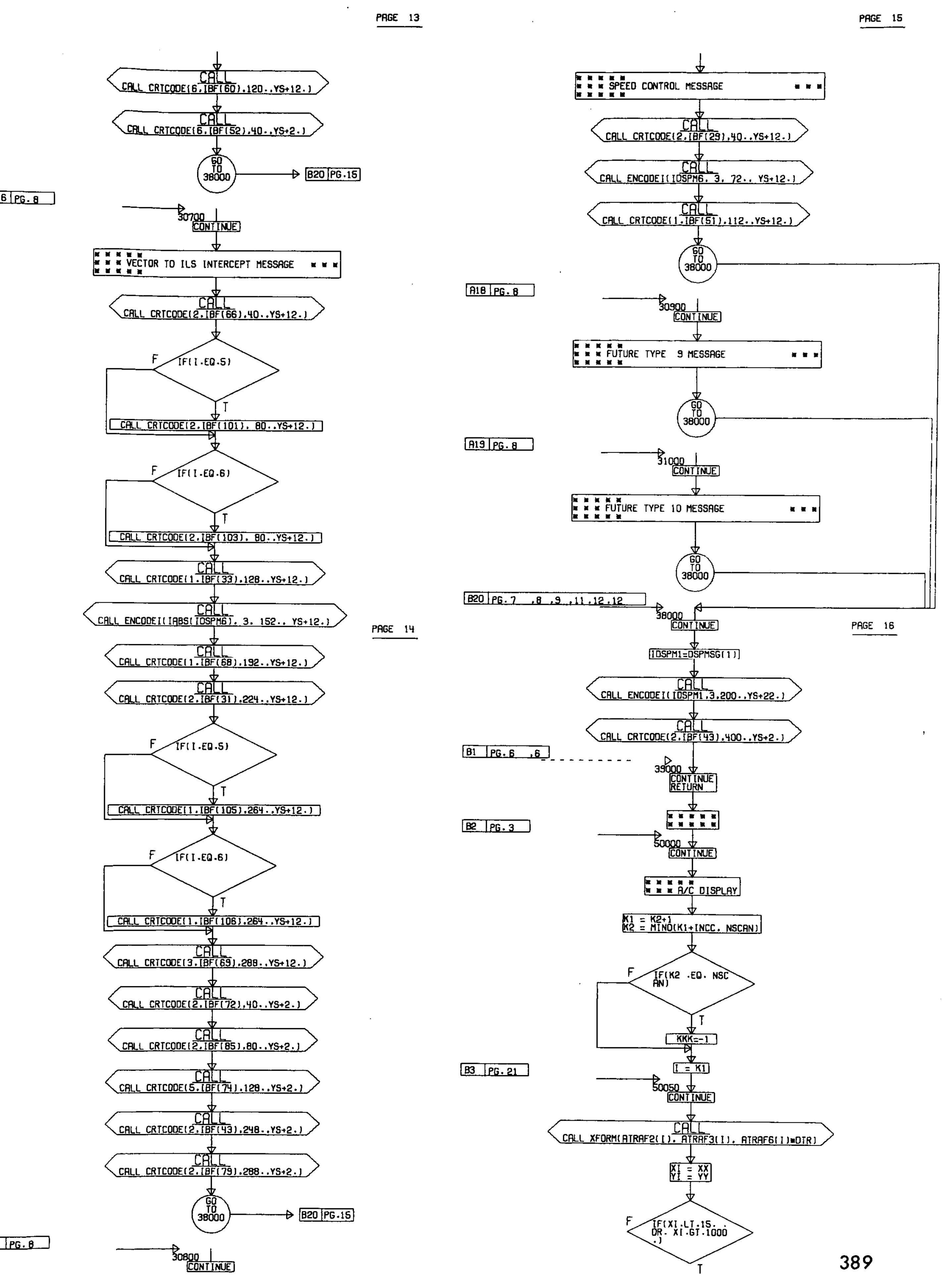
PAGE 4

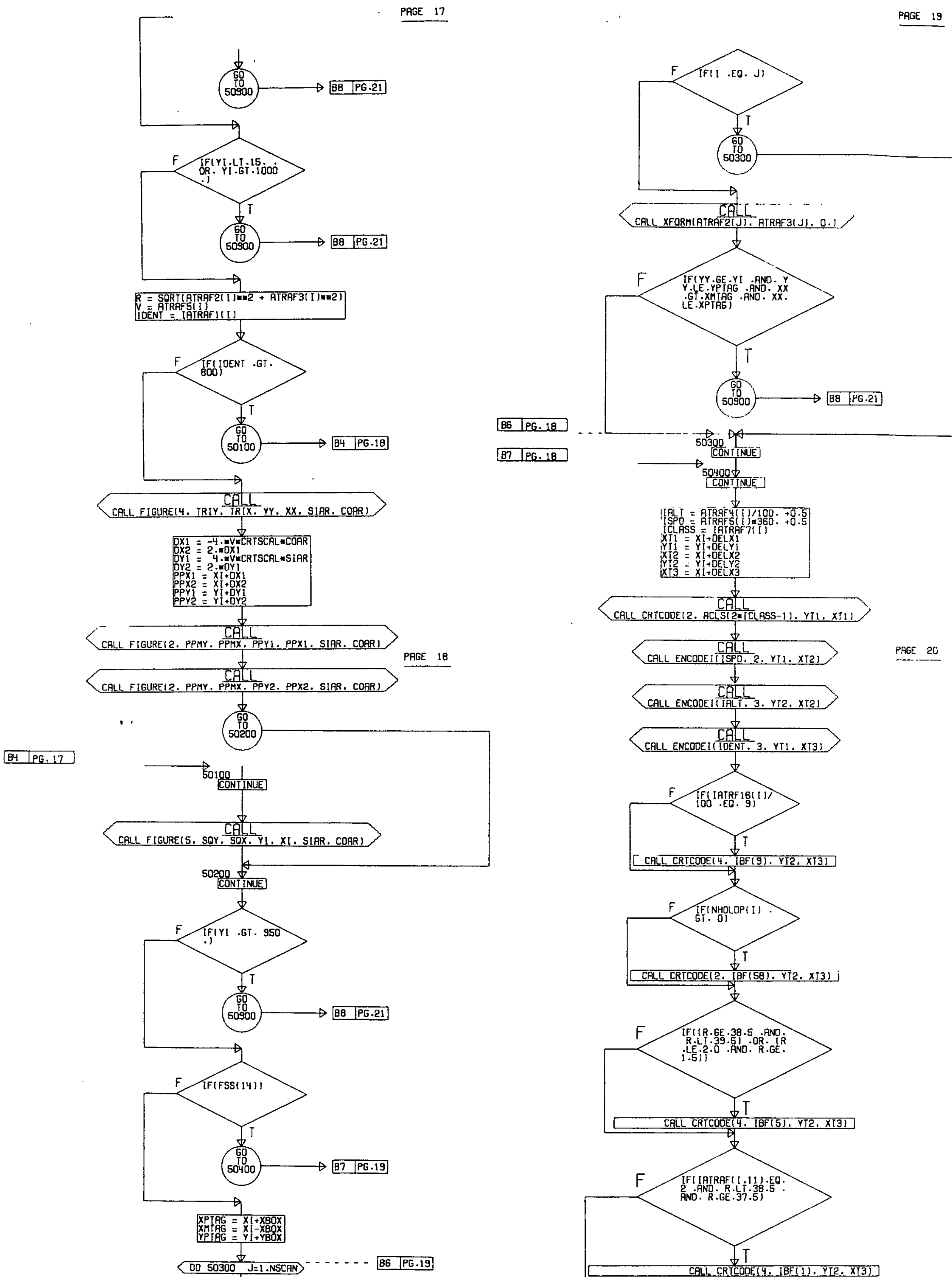
```
CALL
CALL XFORM(XOWN. YOWN. HDOWN*DTR)

IF(XX.LT.15. .
OR. XX.GT.1000
.)
   F
   T  -> GO TO 10100

IF(YY.LT.15. .
OR. YY.GT.1000
.)
   F
   T  -> GO TO 10100

CALL
CALL FIGURE(4. TRIY. TRIX. YY. XX. SIAR. COAR)

A3 PG.3

10100
CONTINUE

IF(IOPT. NE. 3
)
   F
   T  -> GO TO 20000  -> A5 PG.5

CALL
CALL XFORM(0.. 0.. 0.)
```

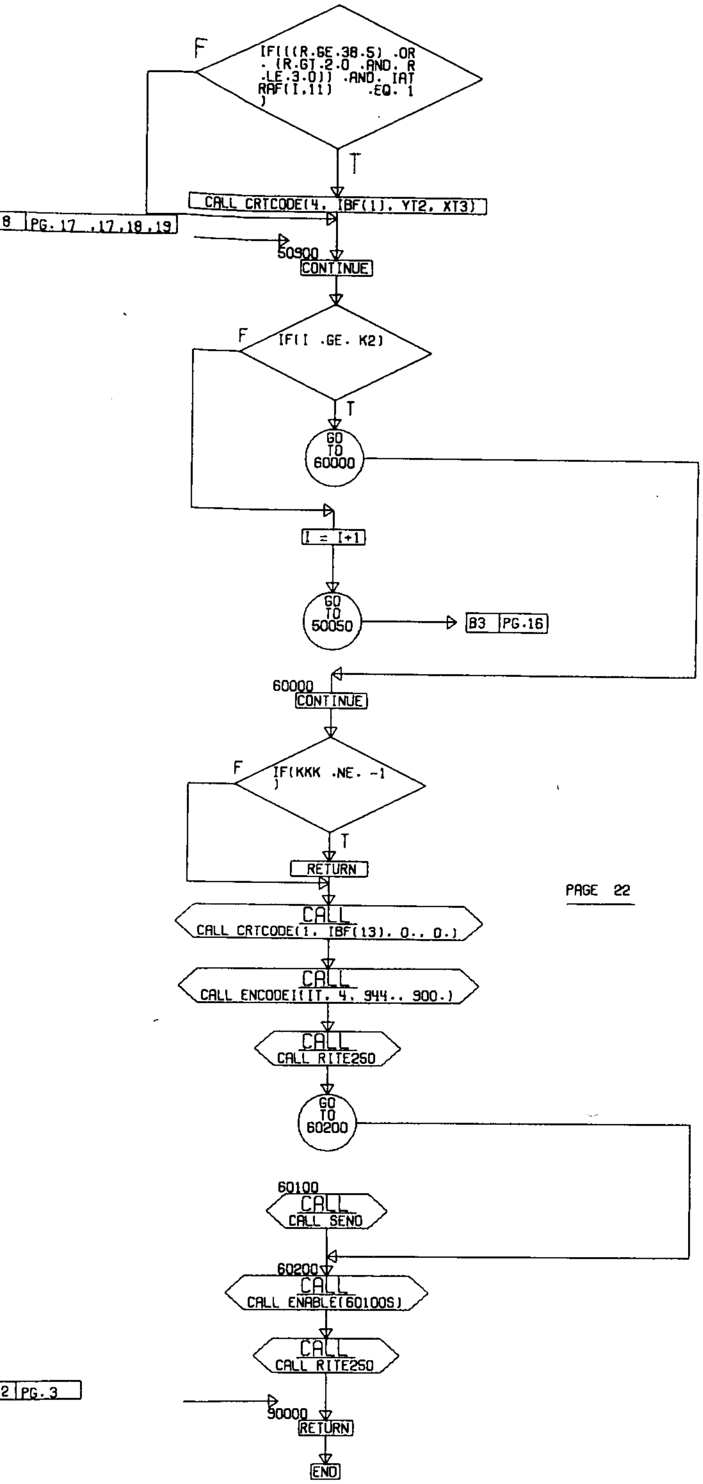


PAGE 5
IF(XX.LT.25. .OR. XX.GT.1000 .)
GO TO 10200
IF(YY.LT.25. .OR. YY.GT.1000 .)
GO TO 10200
CALL FIGURE(1, R9LY, R9LX, YY, XX, SIAR, CORR)
CALL FIGURE(1, R9RY, R9RX, YY, XX, SIAR, CORR)
CALL FIGURE(1, R15Y, R15X, YY, XX, SIAR, CORR)
CALL FIGURE(1, R21Y, R21X, YY, XX, SIAR, CORR)
10200 CONTINUE
CALL ENCODEI(IFIX(PSIREL*RTD+0.5), 3, 482., 515.)
CALL FIGURE(3, YNREF, XNREF, 500., 500., SIAR, CORR)
PAGE 6
PG.4
20000 CONTINUE
KKK = 0
K2 = 0
* * * MESSAGE DISPLAY
CALL CRTCODE(1, IBF(12), 0., 0.)
DO 39000 I=1,10
B1 PG.16
IF(ICOMM(I) .LE. 0)
GO TO 39000
B1 PG.16
ICOMM(I) = ICOMM(I)-1
CALL UPKSPL(IONE,IFIVE,ICMSG2(I,1),IPMSG2,DSPMSG)
CALL UPKSPL(IONE,ITHREE,ICMSG3(I,1),IPMSG3,TEMPA)
DO 30050 J=6,8
JJ = J-5
30050 IDSPMSG(J) = TEMPA(JJ)
MTYPE = DSPMSG(5)
IF(MOD(I,2) .EQ. 0)
PAGE 7
GO TO 30060
YS = 1000-30*(I/2+1)
GO TO 30070
30060 YS = 150-30*I/2
30070 CONTINUE
IF(MTYPE .EQ. 0)
GO TO 38000
B20 PG.15
IF(MTYPE.GT.0 .AND. MTYPE.LT .9)
GO TO 30090
A9 PG.8
PAGE 8
LDISD(115) = .TRUE.
IF(.NOT. FSS(10))
GO TO 38000
B20 PG.15
CALL ABNORML
A9 PG.7
30090 CONTINUE
GO TO(30100, 30200, 30300, 30400, 30500, 30600, 30700, 30800, 30900, 31000) MTYPE
A10 PG.8
A11 PG.9
A12 PG.9
A13 PG.11
A14 PG.11
A15 PG.12
A16 PG.13
A17 PG.14
A18 PG.15
A19 PG.15
A10 PG.8
30100 CONTINUE
* * * DEPARTURE CLEARENCE MESSAGE * * *
IDSPM6 = IDSPM6*100
CALL CRTCODE(8,IBF(20),40.,YS+12.)
CALL ENCODEI(IDSPM6,5,216.,YS+12.)
CALL CRTCODE(3,IBF(28),272.,YS+12.)


PAGE 9
CALL
CALL CRTCODE(10,IBF(31), 40.,YS+ 2.)
CALL
CALL CRTCODE(2,IBF(81),272.,YS+ 2.)
CALL
CALL CRTCODE(2,IBF(41),320.,YS+ 2.)
CALL
CALL CRTCODE(2,IBF(83),360.,YS+ 2.)
GO TO 38000
B20 PG.15
A11 PG.8
30200
CONTINUE
* * * ARRIVAL CLEARENCE MESSAGE * * *
A12 PG.8
30300
CONTINUE
* * * STANDARD HOLD * * *
CALL
CALL CRTCODE(1,IBF(45),40.,YS+12.)
CALL
CALL CRTCODE(2,IBF(34),64.,YS+12.)
F
IF(IDSPM6.EQ.1)
T
CALL CRTCODE(3,IBF(91),168.,YS+12.)
PAGE 10
F
IF(IDSPM6.EQ.2)
T
CALL CRTCODE(2,IBF(94),176.,YS+12.)
F
IF(IDSPM6.EQ.3)
T
CALL CRTCODE(3,IBF(98),176.,YS+12.)
F
IF(IDSPM6.EQ.4)
T
CALL CRTCODE(2,IBF(96),176.,YS+12.)
CALL
CALL ENCODEI(IDSPM7*100,5,304.,YS+12.)
CALL
CALL CRTCODE(3,IBF(48),240.,YS+12.)
CALL
CALL CRTCODE(1,IBF(28),360.,YS+12.)
F
IF(MTYPE.EQ.2)
T
CALL CRTCODE(2,IBF(46),120.,YS+12.)
* * * STANDARD HOLD
PAGE 11
F
IF(MTYPE.EQ.3)
T
CALL CRTCODE(2,IBF(58),128.,YS+12.)
GO TO 38000
B20 PG.15
A13 PG.8
A14 PG.8
30400
CONTINUE
30500
CONTINUE
* * * VECTOR MESSAGE * * *
CALL
CALL CRTCODE(2,IBF(66),40.,YS+12.)
F
IF(IDSPM6.LT.0)
T
CALL CRTCODE(2,IBF(101),80.,YS+12.)
F
IF(IDSPM6.GE.0)
T
CALL CRTCODE(2,IBF(103),80.,YS+12.)
CALL
CALL CRTCODE(1,IBF(33),128.,YS+12.)
PAGE 12
CALL
CALL ENCODEI(IABS(IDSPM6),3,160.,YS+12.)
CALL
CALL CRTCODE(6, IBF(52), 40., YS+2.)
F
IF(MTYPE .EQ. 4)
T
GO TO 38000
B20 PG.15
CALL
CALL ENCODEI(IDSPM8,5,232.,YS+12.)
CALL
CALL CRTCODE(1,IBF(28),280.,YS+12.)
CALL
CALL CRTCODE(1,IBF(50),304.,YS+12.)
CALL
CALL ENCODEI(IDSPM7,3,328.,YS+12.)
CALL
CALL CRTCODE(1,IBF(51),360.,YS+12.)
GO TO 38000
B20 PG.15
A15 PG.8
30600
CONTINUE
* * * EMERGENCY HOLD MESSAGE * * *
CALL
CALL CRTCODE(2,IBF(58),40.,YS+12.)
CALL
CALL CRTCODE(2,IBF(43),80.,YS+12.)

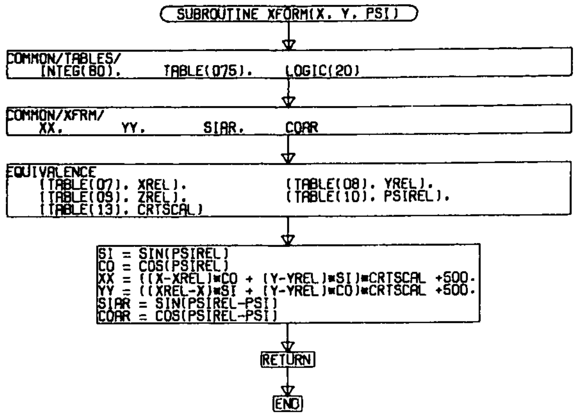

PAGE 13
CALL CRTCODE(6.IBF(60).120..YS+12.)
CALL CRTCODE(6.IBF(52).40..YS+2.)
GO TO 38000
B20 PG.15
6 PG.8
30700 CONTINUE
VECTOR TO ILS INTERCEPT MESSAGE
CALL CRTCODE(2.IBF(66).40..YS+12.)
IF(I.EQ.5)
CALL CRTCODE(2.IBF(101). 80..YS+12.)
IF(I.EQ.6)
CALL CRTCODE(2.IBF(103). 80..YS+12.)
CALL CRTCODE(1.IBF(33).128..YS+12.)
CALL ENCODEI(IABS(IDSPM6). 3. 152.. YS+12.)
PAGE 14
CALL CRTCODE(1.IBF(68).192..YS+12.)
CALL CRTCODE(2.IBF(31).224..YS+12.)
IF(I.EQ.5)
CALL CRTCODE(1.IBF(105).264..YS+12.)
IF(I.EQ.6)
CALL CRTCODE(1.IBF(106).264..YS+12.)
CALL CRTCODE(3.IBF(69).288..YS+12.)
CALL CRTCODE(2.IBF(72).40..YS+2.)
CALL CRTCODE(2.IBF(85).80..YS+2.)
CALL CRTCODE(5.IBF(74).128..YS+2.)
CALL CRTCODE(2.IBF(43).248..YS+2.)
CALL CRTCODE(2.IBF(79).288..YS+2.)
GO TO 38000
B20 PG.15
PG.8
30800 CONTINUE
PAGE 15
SPEED CONTROL MESSAGE
CALL CRTCODE(2.IBF(29).40..YS+12.)
CALL ENCODEI(IDSPM6. 3. 72.. YS+12.)
CALL CRTCODE(1.IBF(51).112..YS+12.)
GO TO 38000
A18 PG.8
30900 CONTINUE
FUTURE TYPE 9 MESSAGE
GO TO 38000
A19 PG.8
31000 CONTINUE
FUTURE TYPE 10 MESSAGE
GO TO 38000
B20 PG.7 ,8 ,9 ,11 ,12 ,12
38000 CONTINUE
PAGE 16
IDSPM1=DSPMSG(1)
CALL ENCODEI(IDSPM1.3.200..YS+22.)
CALL CRTCODE(2.IBF(43).400..YS+2.)
B1 PG.6 ,6
39000 CONTINUE RETURN
B2 PG.3
50000 CONTINUE
A/C DISPLAY
K1 = K2+1
K2 = MIN0(K1+INCC. NSCAN)
IF(K2 .EQ. NSCAN)
KKK=-1
I = K1
B3 PG.21
50050 CONTINUE
CALL XFORM(ATRAF2(I). ATRAF3(I). ATRAF6(I)*DTR)
XI = XX
YI = YY
IF(XI.LT.15. .OR. XI.GT.1000.)

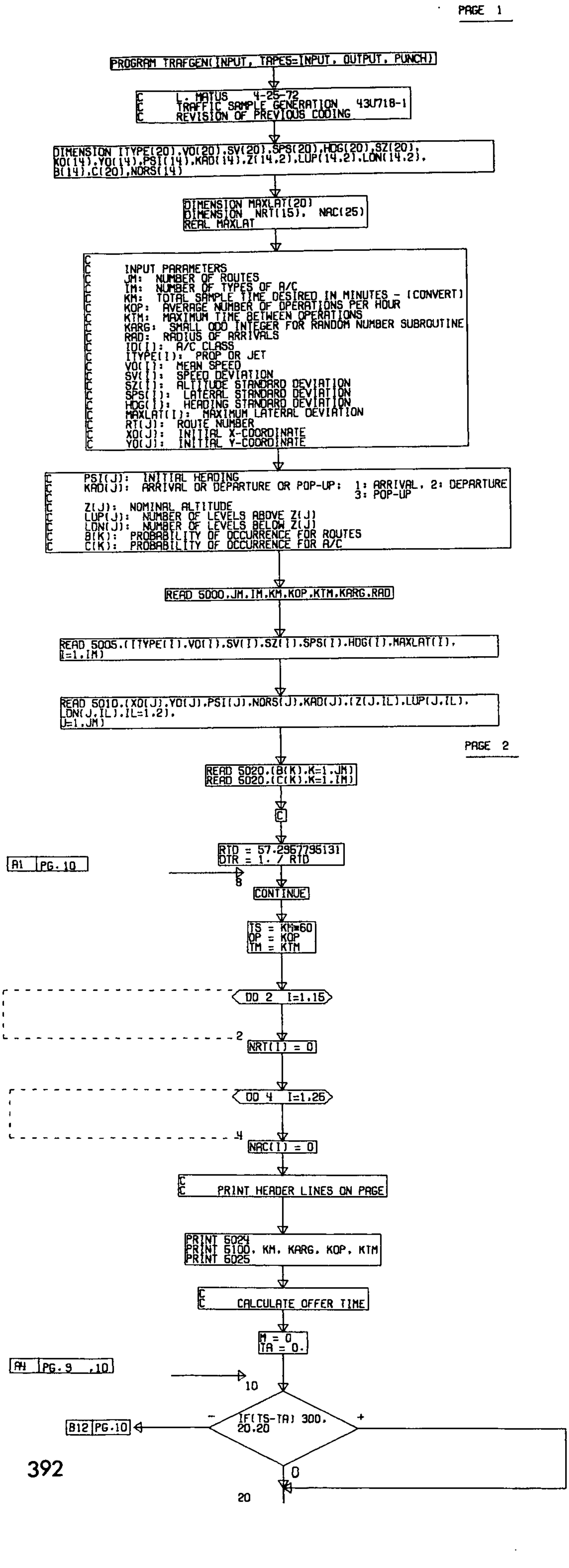

PAGE 17
GO TO 50900
B8 PG.21
IF(YI.LT.15. .OR. YI.GT.1000 .)
F
T
GO TO 50900
B8 PG.21
R = SQRT(ATRAF2(I)**2 + ATRAF3(I)**2)
V = ATRAF5(I)
IDENT = IATRAF1(I)
IF(IDENT .GT. 800)
F
T
GO TO 50100
B4 PG.18
CALL
CALL FIGURE(4. TRIY. TRIX. YY. XX. SIAR. COAR)
DX1 = -4.*V*CRTSCAL*COAR
DX2 = 2.*DX1
DY1 = 4.*V*CRTSCAL*SIAR
DY2 = 2.*DY1
PPX1 = XI+DX1
PPX2 = XI+DX2
PPY1 = YI+DY1
PPY2 = YI+DY2
CALL
CALL FIGURE(2. PPMY. PPMX. PPY1. PPX1. SIAR. COAR)
PAGE 18
CALL
CALL FIGURE(2. PPMY. PPMX. PPY2. PPX2. SIAR. COAR)
GO TO 50200
B4 PG.17
50100
CONTINUE
CALL
CALL FIGURE(5. SQY. SQX. YI. XI. SIAR. COAR)
50200
CONTINUE
IF(YI .GT. 950 .)
F
T
GO TO 50900
B8 PG.21
IF(FSS(14))
F
T
GO TO 50400
B7 PG.19
XPTAG = XI+XBOX
XMTAG = XI-XBOX
YPTAG = YI+YBOX
DO 50300 J=1.NSCAN
B6 PG.19
PAGE 19
IF(I .EQ. J)
F
T
GO TO 50300
CALL
CALL XFORM(ATRAF2(J). ATRAF3(J). 0.)
IF(YY.GE.YI .AND. YY.LE.YPTAG .AND. XX.GT.XMTAG .AND. XX.LE.XPTAG)
F
T
GO TO 50900
B8 PG.21
B6 PG.18
B7 PG.18
50300
CONTINUE
50400
CONTINUE
IALT = ATRAF4(I)/100. +0.5
ISPD = ATRAF5(I)*360. +0.5
ICLASS = IATRAF7(I)
XT1 = XI+DELX1
YT1 = YI+DELY1
XT2 = XI+DELX2
YT2 = YI+DELY2
XT3 = XI+DELX3
CALL
CALL CRTCODE(2. ACLS(2*ICLASS-1). YT1. XT1)
PAGE 20
CALL
CALL ENCODEI(ISPD. 2. YT1. XT2)
CALL
CALL ENCODEI(IALT. 3. YT2. XT2)
CALL
CALL ENCODEI(IDENT. 3. YT1. XT3)
IF(IATRF16(I)/100 .EQ. 9)
F
T
CALL CRTCODE(4. IBF(9). YT2. XT3)
IF(NHOLDP(I) .GT. 0)
F
T
CALL CRTCODE(2. IBF(58). YT2. XT3)
IF((R.GE.38.5 .AND. R.LT.39.5) .OR. (R .LE.2.0 .AND. R.GE. 1.5))
F
T
CALL CRTCODE(4. IBF(5). YT2. XT3)
IF(IATRAF(I.11).EQ. 2 .AND. R.LT.38.5 . AND. R.GE.37.5)
F
T
CALL CRTCODE(4. IBF(1). YT2. XT3)


IF(((R.GE.38.5) .OR. (R.GT.2.0 .AND. R .LE.3.0)) .AND. IATRAF(I,11) .EQ. 1)
F
T
CALL CRTCODE(4, IBF(1), YT2, XT3)
8 PG.17 ,17,18,19
50900
CONTINUE
IF(I .GE. K2)
F
T
GO TO 60000
I = I+1
GO TO 50050
B3 PG.16
60000
CONTINUE
IF(KKK .NE. -1)
F
T
RETURN
CALL
CALL CRTCODE(1, IBF(13), 0., 0.)
CALL
CALL ENCODEI(IT, 4, 944., 900.)
CALL
CALL RITE250
GO TO 60200
60100
CALL
CALL SEND
60200
CALL
CALL ENABLE(60100S)
CALL
CALL RITE250
2 PG.3
90000
RETURN
END

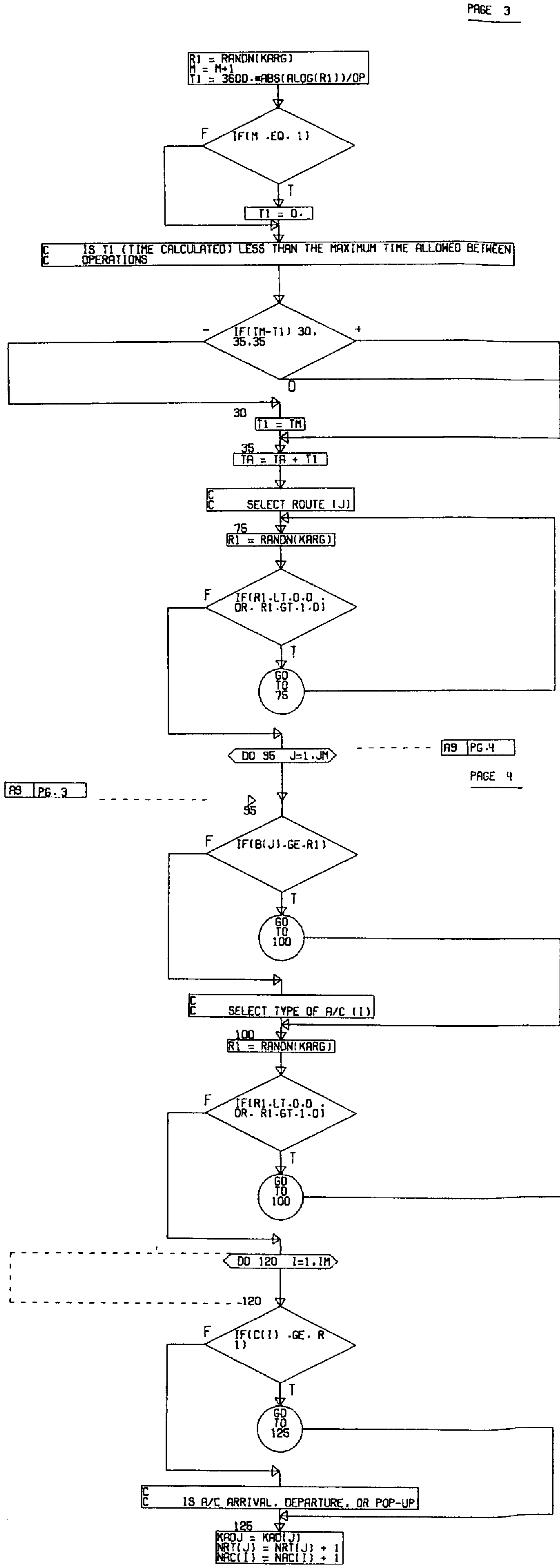
PAGE 22

NO. 1

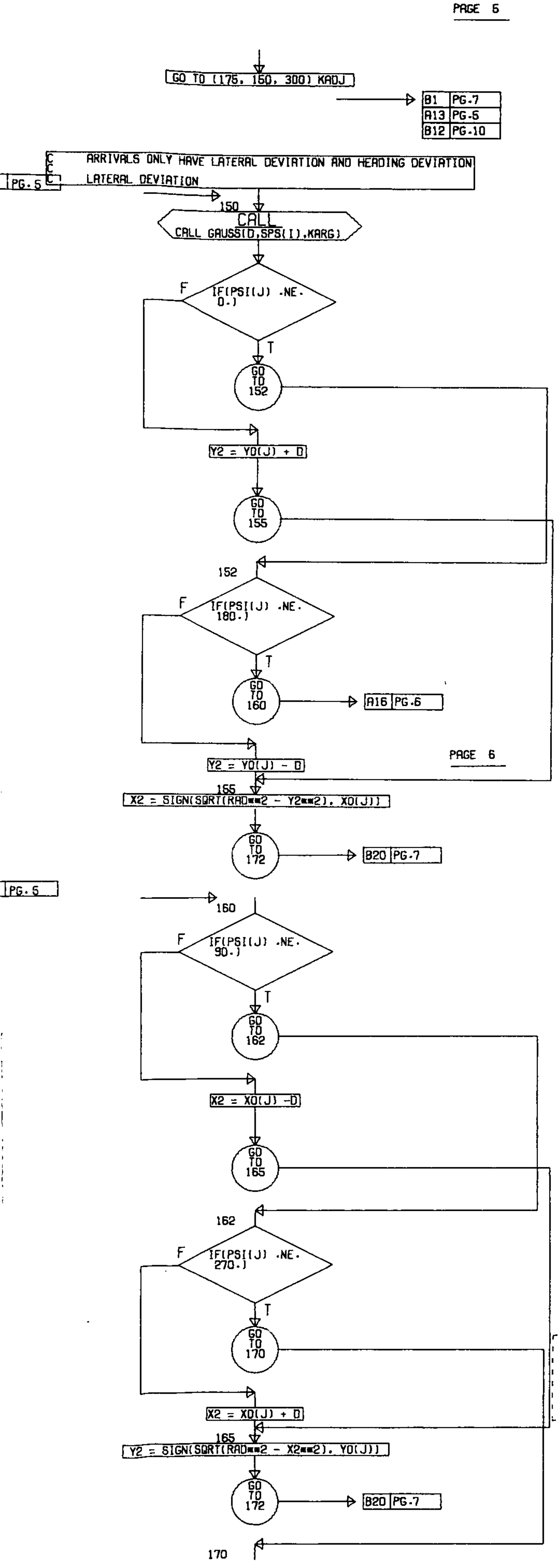

SUBROUTINE XFORM(X, Y, PSI)
COMMON/TABLES/
INTEG(80), TABLE(075), LOGIC(20)
COMMON/XFRM/
XX, YY, SIAR, COAR
EQUIVALENCE
(TABLE(07), XREL), (TABLE(08), YREL),
(TABLE(09), ZREL), (TABLE(10), PSIREL),
(TABLE(13), CRTSCAL)
SI = SIN(PSIREL)
CO = COS(PSIREL)
XX = ((X-XREL)*CO + (Y-YREL)*SI)*CRTSCAL +500.
YY = ((XREL-X)*SI + (Y-YREL)*CO)*CRTSCAL +500.
SIAR = SIN(PSIREL-PSI)
COAR = COS(PSIREL-PSI)
RETURN
END

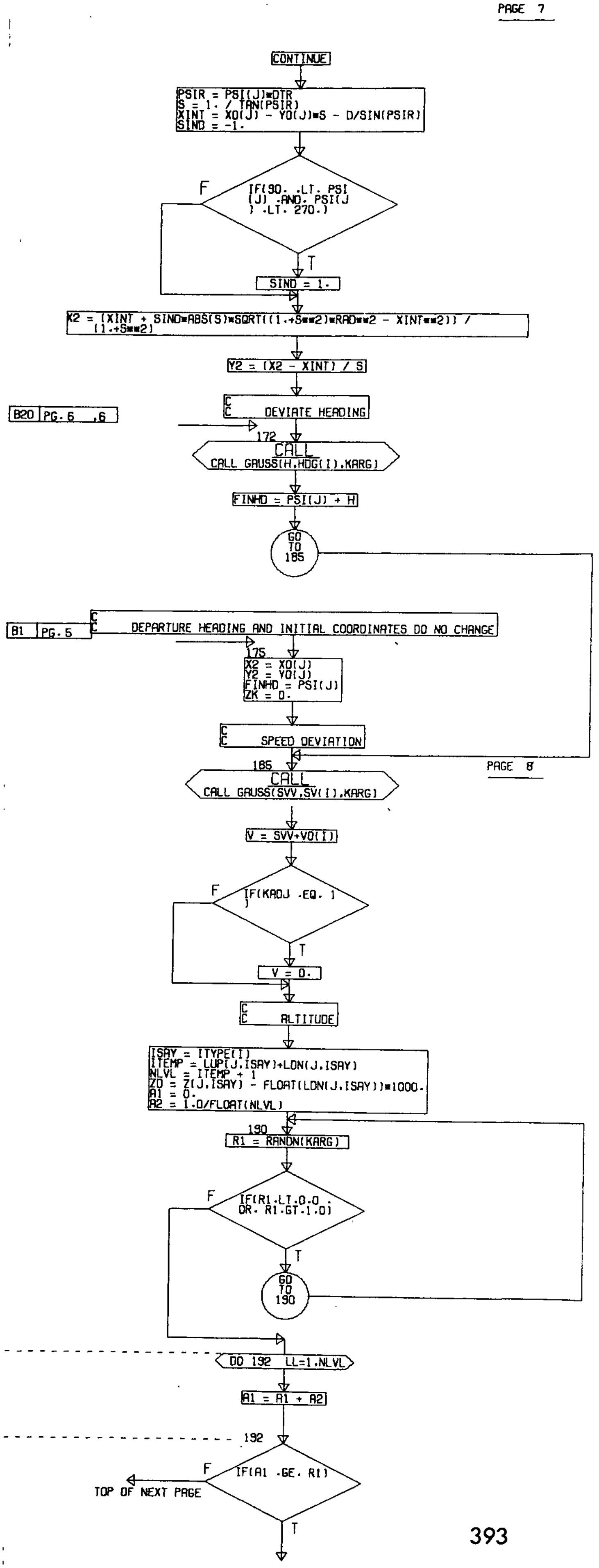


PAGE 1
PROGRAM TRAFGEN(INPUT, TAPES=INPUT, OUTPUT, PUNCH)
C L. MATUS 4-25-72
C TRAFFIC SAMPLE GENERATION 43U718-1
C REVISION OF PREVIOUS CODING
DIMENSION ITYPE(20),VO(20),SV(20),SPS(20),HDG(20),SZ(20),
XO(14),YO(14),PSI(14),KAD(14),Z(14,2),LUP(14,2),LDN(14,2),
B(14),C(20),NORS(14)
DIMENSION MAXLAT(20)
DIMENSION NRT(15), NAC(25)
REAL MAXLAT
C INPUT PARAMETERS
C JM: NUMBER OF ROUTES
C IM: NUMBER OF TYPES OF A/C
C KM: TOTAL SAMPLE TIME DESIRED IN MINUTES - (CONVERT)
C KOP: AVERAGE NUMBER OF OPERATIONS PER HOUR
C KTM: MAXIMUM TIME BETWEEN OPERATIONS
C KARG: SMALL ODD INTEGER FOR RANDOM NUMBER SUBROUTINE
C RAD: RADIUS OF ARRIVALS
C ID(I): A/C CLASS
C ITYPE(I): PROP OR JET
C VO(I): MEAN SPEED
C SV(I): SPEED DEVIATION
C SZ(I): ALTITUDE STANDARD DEVIATION
C SPS(I): LATERAL STANDARD DEVIATION
C HDG(I): HEADING STANDARD DEVIATION
C MAXLAT(I): MAXIMUM LATERAL DEVIATION
C RT(J): ROUTE NUMBER
C XO(J): INITIAL X-COORDINATE
C YO(J): INITIAL Y-COORDINATE
C PSI(J): INITIAL HEADING
C KAD(J): ARRIVAL OR DEPARTURE OR POP-UP: 1: ARRIVAL, 2: DEPARTURE
3: POP-UP
C Z(J): NOMINAL ALTITUDE
C LUP(J): NUMBER OF LEVELS ABOVE Z(J)
C LDN(J): NUMBER OF LEVELS BELOW Z(J)
C B(K): PROBABILITY OF OCCURRENCE FOR ROUTES
C C(K): PROBABILITY OF OCCURRENCE FOR A/C
READ 5000,JM,IM,KM,KOP,KTM,KARG,RAD
READ 5005,(ITYPE(I),VO(I),SV(I),SZ(I),SPS(I),HDG(I),MAXLAT(I),
I=1,IM)
READ 5010,(XO(J),YO(J),PSI(J),NORS(J),KAD(J),(Z(J,IL),LUP(J,IL),
LDN(J,IL),IL=1,2),
J=1,JM)
PAGE 2
READ 5020,(B(K),K=1,JM)
READ 5020,(C(K),K=1,IM)
C
RTD = 57.2957795131
DTR = 1. / RTD
A1 PG.10
8
CONTINUE
TS = KM*60
OP = KOP
TM = KTM
DO 2 I=1,15
2
NRT(I) = 0
DO 4 I=1,25
4
NAC(I) = 0
C PRINT HEADER LINES ON PAGE
PRINT 6024
PRINT 6100, KM, KARG, KOP, KTM
PRINT 6025
C CALCULATE OFFER TIME
M = 0
TA = 0.
A4 PG.9 ,10
10
B12 PG.10
-
IF(TS-TA) 300,
20,20
+
0
20

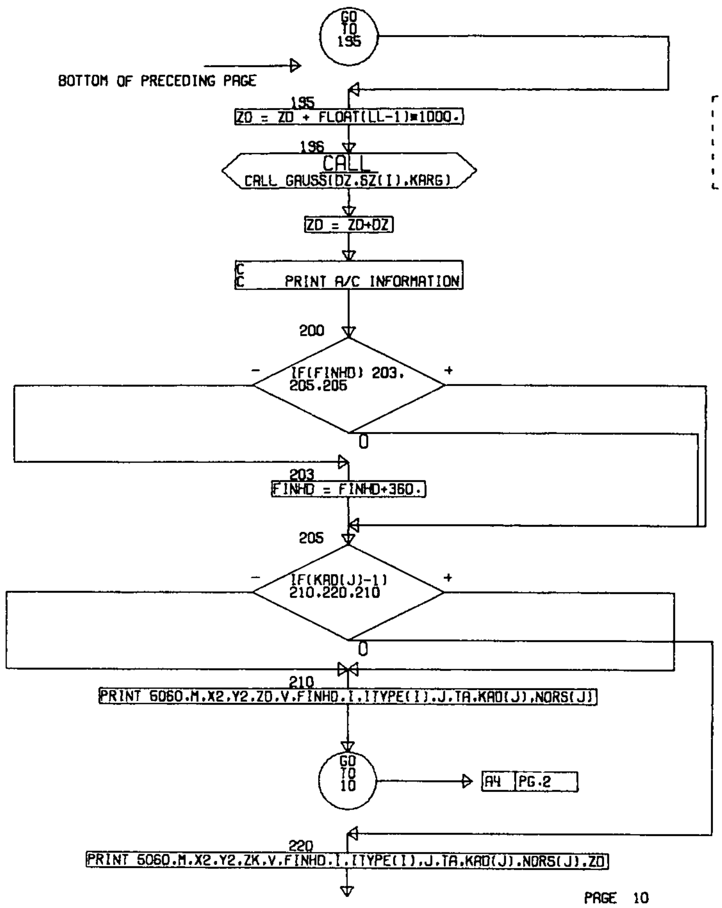


PAGE 3
R1 = RANDN(KARG)
M = M+1
T1 = 3600.*ABS(ALOG(R1))/OP
F
IF(M .EQ. 1)
T
T1 = 0.
C IS T1 (TIME CALCULATED) LESS THAN THE MAXIMUM TIME ALLOWED BETWEEN
C OPERATIONS
-
IF(TM-T1) 30,
35,35
+
0
30
T1 = TM
35
TA = TA + T1
C SELECT ROUTE (J)
75
R1 = RANDN(KARG)
F
IF(R1.LT.0.0 .
OR. R1.GT.1.0)
T
GO TO 75
DO 95 J=1,JM
A9 PG.4
PAGE 4
A9 PG.3
95
F
IF(B(J).GE.R1)
T
GO TO 100
C SELECT TYPE OF A/C (I)
100
R1 = RANDN(KARG)
F
IF(R1.LT.0.0 .
OR. R1.GT.1.0)
T
GO TO 100
DO 120 I=1,IM
120
F
IF(C(I) .GE. R
1)
T
GO TO 125
C IS A/C ARRIVAL, DEPARTURE, OR POP-UP
125
KADJ = KAD(J)
NRT(J) = NRT(J) + 1
NAC(I) = NAC(I) + 1

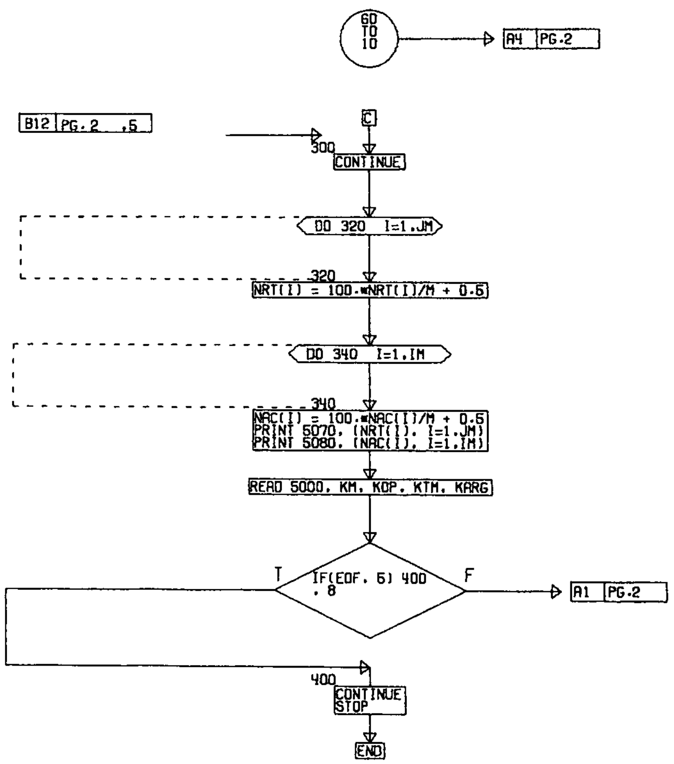

PAGE 6
GO TO (175, 150, 300) KADJ
B1 PG.7
A13 PG.5
B12 PG.10
C ARRIVALS ONLY HAVE LATERAL DEVIATION AND HEADING DEVIATION
C
C LATERAL DEVIATION
PG.5
150
CALL
CALL GAUSS(D,SPS(I),KARG)
IF(PSI(J) .NE. 0.)
F
T
GO TO 152
Y2 = YO(J) + D
GO TO 155
152
IF(PSI(J) .NE. 180.)
GO TO 160
A16 PG.6
Y2 = YO(J) - D
PAGE 6
155
X2 = SIGN(SQRT(RAD**2 - Y2**2), XO(J))
GO TO 172
B20 PG.7
PG.5
160
IF(PSI(J) .NE. 90.)
GO TO 162
X2 = XO(J) -D
GO TO 165
162
IF(PSI(J) .NE. 270.)
GO TO 170
X2 = XO(J) + D
165
Y2 = SIGN(SQRT(RAD**2 - X2**2), YO(J))
GO TO 172
B20 PG.7
170


PAGE 7
CONTINUE
PSIR = PSI(J)*DTR
S = 1. / TAN(PSIR)
XINT = XO(J) - YO(J)*S - D/SIN(PSIR)
SIND = -1.
IF(90. .LT. PSI(J) .AND. PSI(J) .LT. 270.)
F
T
SIND = 1.
X2 = (XINT + SIND*ABS(S)*SQRT((1.+S**2)*RAD**2 - XINT**2)) / (1.+S**2)
Y2 = (X2 - XINT) / S
C
C DEVIATE HEADING
B20 PG.6 ,6
172
CALL
CALL GAUSS(H,HDG(I),KARG)
FINHD = PSI(J) + H
GO TO 185
B1 PG.5
C
C DEPARTURE HEADING AND INITIAL COORDINATES DO NO CHANGE
175
X2 = XO(J)
Y2 = YO(J)
FINHD = PSI(J)
ZK = 0.
C
C SPEED DEVIATION
PAGE 8
185
CALL
CALL GAUSS(SVV,SV(I),KARG)
V = SVV+VO(I)
IF(KADJ .EQ. 1)
V = 0.
C
C ALTITUDE
ISAY = ITYPE(I)
ITEMP = LUP(J,ISAY)+LDN(J,ISAY)
NLVL = ITEMP + 1
ZO = Z(J,ISAY) - FLOAT(LDN(J,ISAY))*1000.
A1 = 0.
A2 = 1.0/FLOAT(NLVL)
190
R1 = RANDN(KARG)
IF(R1.LT.0.0 .OR. R1.GT.1.0)
GO TO 190
DO 192 LL=1,NLVL
A1 = A1 + A2
192
IF(A1 .GE. R1)
TOP OF NEXT PAGE

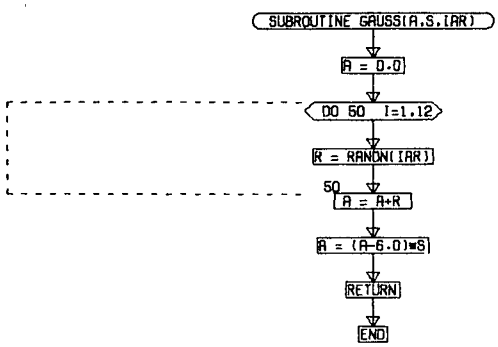
PAGE 9



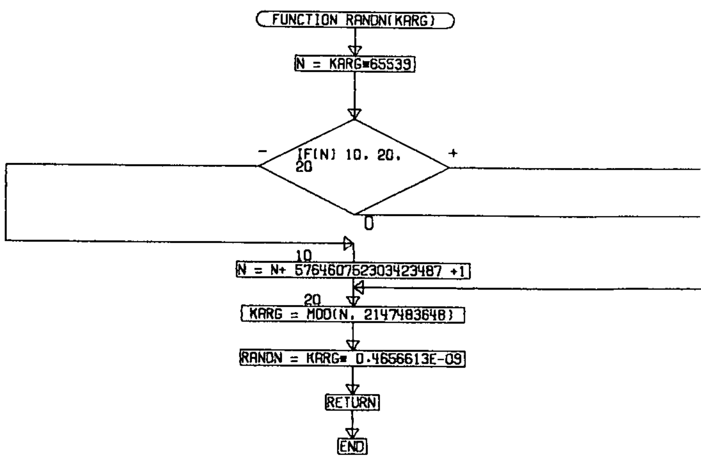
PAGE 10



PAGE 11



PAGE 12



PAGE 1

```
PROGRAM ANALYS(INPUT, OUTPUT,TAPE5)

* * * MAIN, RTAPE, ST4RR

COMMON/BLK1/
     ITRKNO(50)

* * * MAIN, RTAPE

COMMON/BLK2/
     IDEL(50),       IRUNTP,         ITTP,
     NDEL,           XRWY(2),        YRWY(2)
   , ETAPE

LOGICAL ETAPE

* * * MAIN, ST4RR, RTAPE

COMMON/BLK3/
     ITRKB1(50),     NK,             NTRKB1

* * * MAIN, OUTPT1

COMMON/BLK4/
     DELT,           IKNT(2)

* * * MAIN, RTAPE, OUTPT1, ORDER

COMMON/BLK5/
     NEVENT(500,4,2)

* * * ST4RR, MAIN, OUTST4

COMMON/BLK8/
     BDP(40),        IHISTP(40,5),
     KCOUNT(5),      NCATP,          NCPM1,
     NKM4

* * * RTAPE, MAIN, FILT4R

COMMON/BLK9/
     DATA(50,8)

* * * MAIN, FILT4R
```

PAGE 2

```
COMMON/BLK10/
     FILFAC(5),      INDFIL

* * * MAIN, ORDUP

COMMON/BLK11/
     IDIR

* * * MAIN, OUTST4

COMMON/BLK12/
     IHEAD(8),       IQ,             IR,
     KHEAD(4,3),     NCTOT

* * * RTAPE, ST4RR, MAIN, ORDUP, FILT4R

COMMON/BLK13/
     NCRAFT

* * * ST4RR, MAIN, ORDUP, FILT4R

COMMON/BLK14/
     WORK(50,50)

* * * ST4RR, MAIN, FILT4R

COMMON/BLK15/
     VC(50, 50, 2)

* * * MAIN, OUTST4, OUTPUT

COMMON/BLK18/
     IRUN,           ISTART,         ISTOP

* * * MAIN, FILT4R, OUTST4

COMMON/BLK19/
     IPARM

DIMENSION
     IDATA(50,8),    IEVENT(500),                RELR(50,50)

EQUIVALENCE
     (DATA, IDATA),                  (RELR, WORK)

DELT = 4.
ETAPE = .FALSE.
ITAPE = 5
U = 16.1/6076.1
```

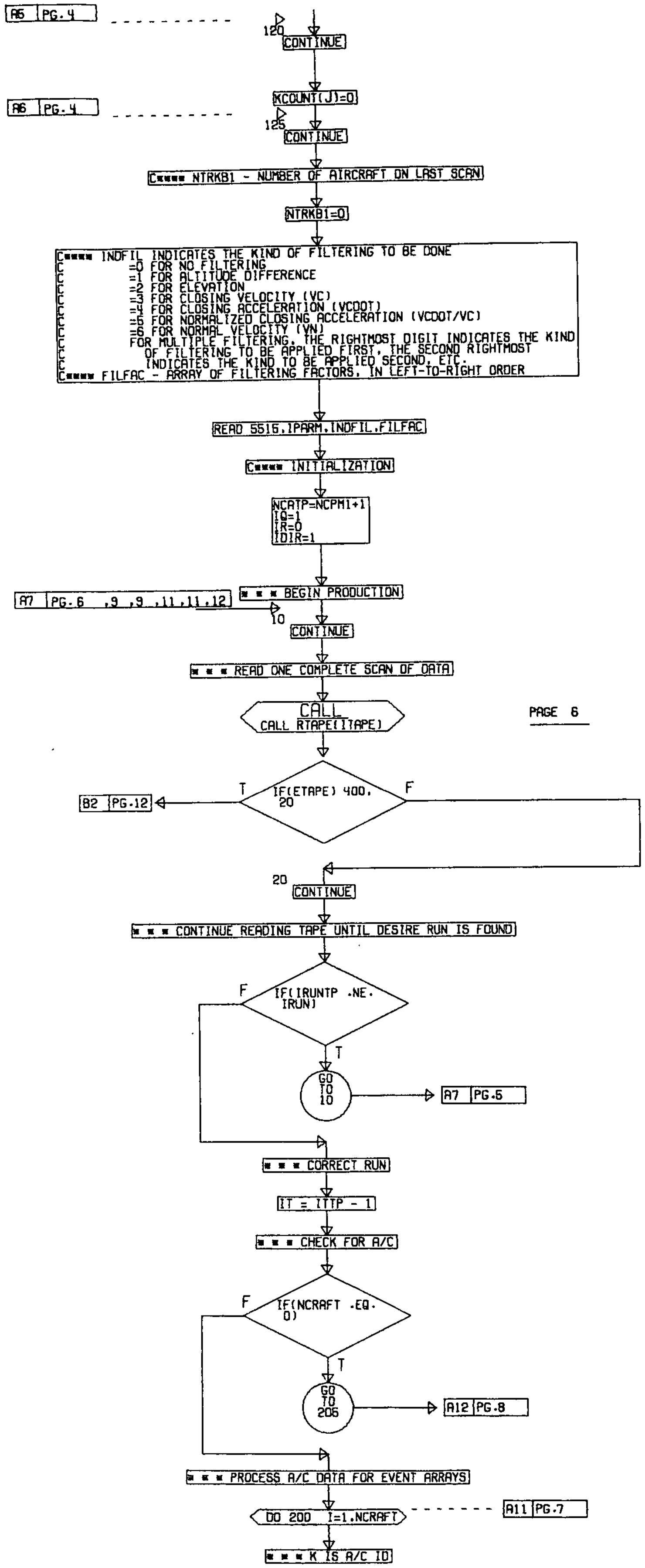
PAGE 3

```
DO 3  I=1,500
3   IEVENT(I) = 0
DO 4  K=1,2
IKNT(K) =0
DO 4  J=1,4
DO 4  I=1,500
4   NEVENT(I,J,K) = 0
DO 5  I=1,50
DO 6  J=1,50
RELR(I,J) = 0.
DO 5  K=1,2
5   VC(I,J,K) = 0.
NCTOT = 0
II = 0
ITTP = 1

* * * READ RUNWAY TOUCHDOWN COORDINATES

READ 5000, (XRWY(I), YRWY(I), I=1,2)

* * * READ DESIRED RUN NO., FIRST AND LAST SCANS DESIRED
```

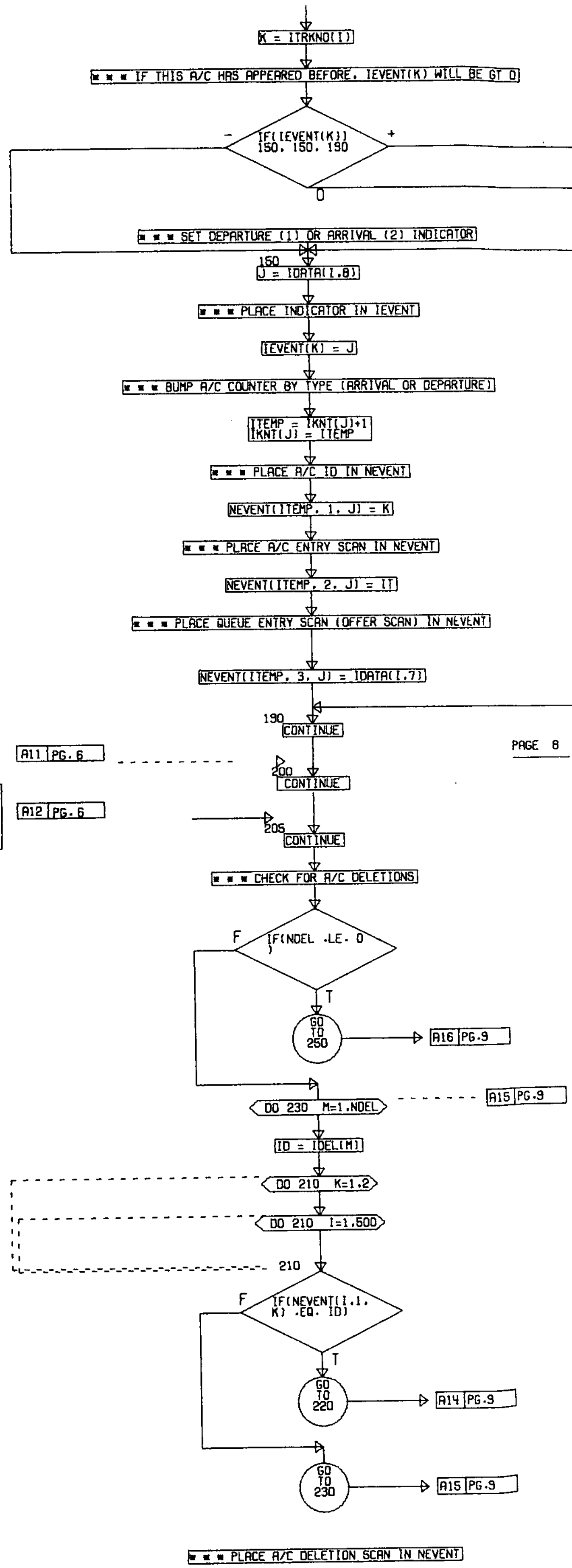
PAGE 4

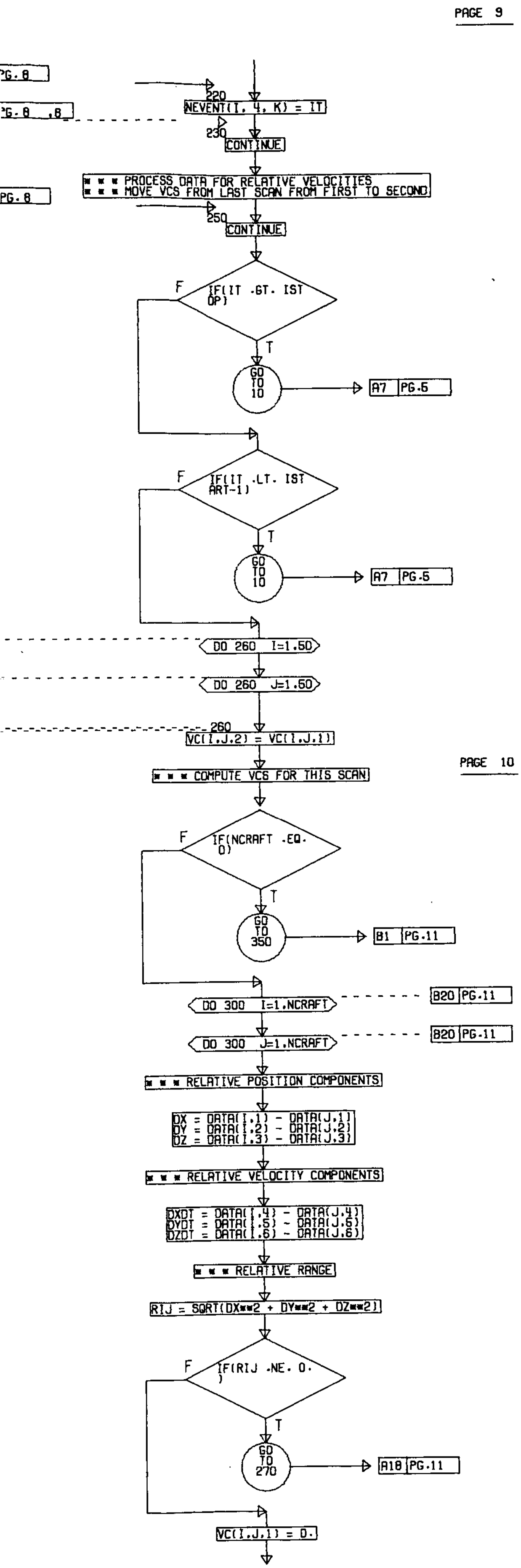
```
READ 5010, IRUN, ISTART, ISTOP

C**** NK - INDICATES THE LAST POSITION FOR WHICH HISTOGRAMS ARE TO
C****      BE COMPUTED - FIVE POSITIONS ARE COMPUTED EACH TIME, THEREFORE
C****      FOR EXAMPLE, IF NK=10 THE HISTOGRAMS FOR RELATIVE RANGE WILL
C****      BE FOR THE 6TH THROUGH THE 10TH CLOSEST AIRCRAFT PAIRS
C**** NCPM1 - THE NUMBER OF CATEGORY BOUNDARIES (ONE LESS THAN THE
C****      NUMBER OF CATEGORIES) IN THE PERCENT-TIME HISTOGRAM

READ 5010,NK,NCPM1

C**** KHEAD - ARRAY CONTAINING THE NAME OF THE DEFINING PARAMETER

READ 5500,KHEAD

C**** BDP - ARRAY OF CATEGORY BOUNDARIES

READ 5510,(BDP(I),I=1,NCPM1)

C**** IHEAD - ARRAY CONTAINING NAME OF FILTERING PARAMETER

READ 5500,(IHEAD(I),I=1,8)

C**** IPARM - IDENTIFIES THE DEFINING PARAMETER
C****      =1 FOR RELATIVE RANGE
C****      =2 FOR TAU
C****      =3 FOR MODIFIED TAU

NKM4=NK-4

DO 100  I=1,50

C**** ITRKB1 - AIRCRAFT NUMBERS ON LAST SCAN

ITRKB1(I)=0
100 ITRKNO(I)=0

C**** INITIALIZE HISTOGRAM

DO 126  J=1,5          - - - - - -  A6 PG.5
DO 120  I=1,40         - - - - - -  A5 PG.5
IHISTP(I,J)=0
```

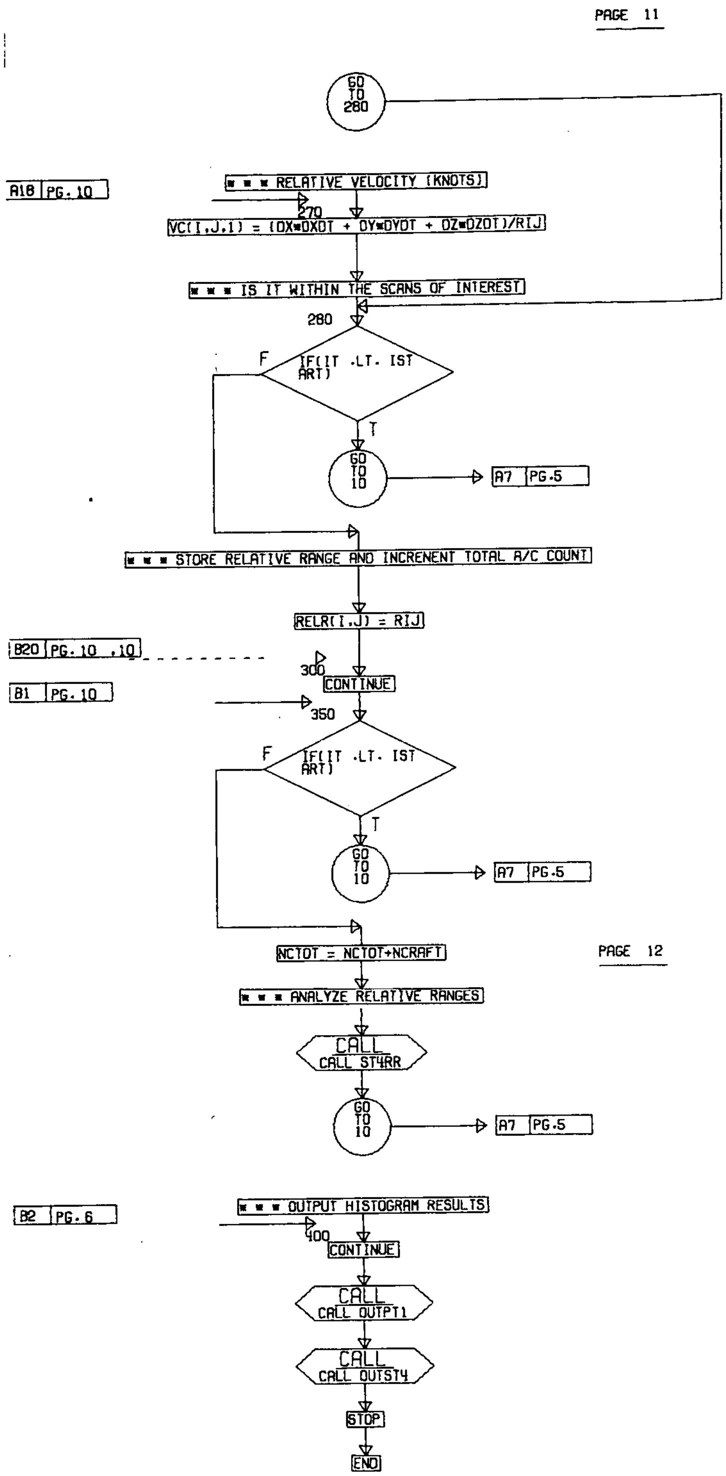
PAGE 6

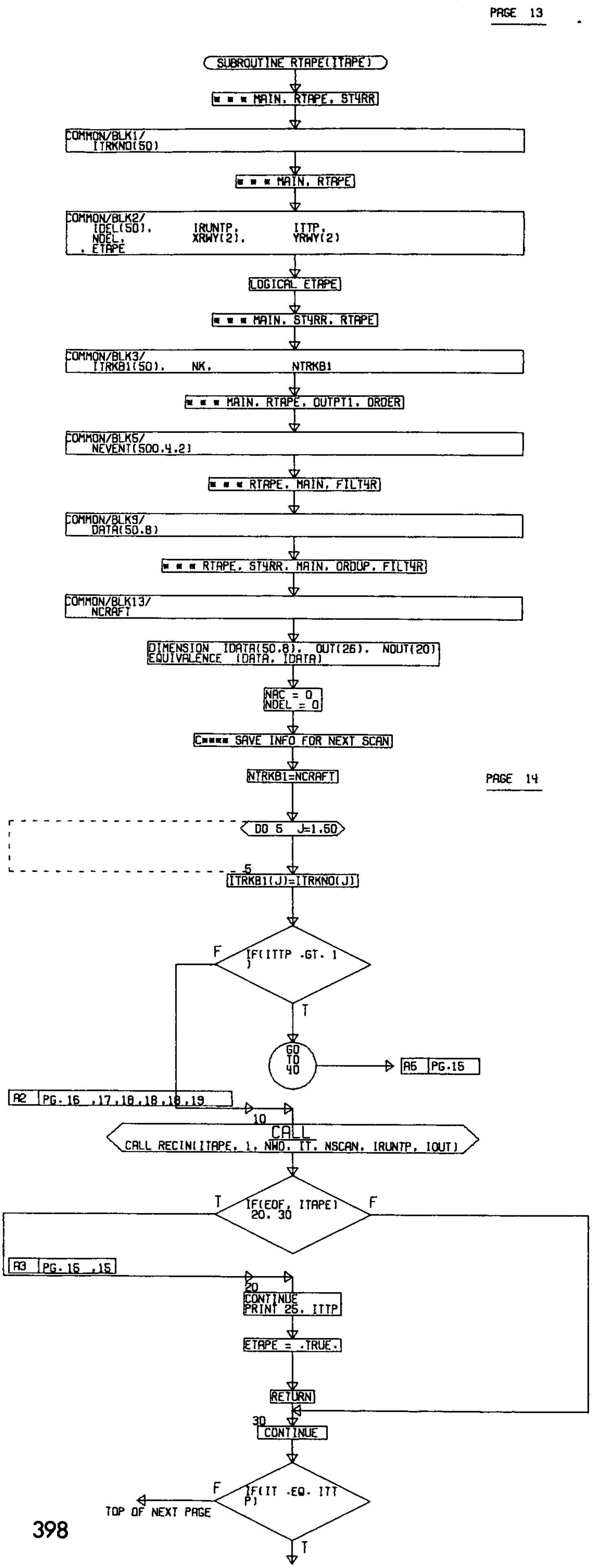

A5 PG.4
120 CONTINUE
KCOUNT(J)=0
A6 PG.4
125 CONTINUE
C***** NTRKB1 - NUMBER OF AIRCRAFT ON LAST SCAN
NTRKB1=0
C***** INDFIL INDICATES THE KIND OF FILTERING TO BE DONE
C =0 FOR NO FILTERING
C =1 FOR ALTITUDE DIFFERENCE
C =2 FOR ELEVATION
C =3 FOR CLOSING VELOCITY (VC)
C =4 FOR CLOSING ACCELERATION (VCDOT)
C =5 FOR NORMALIZED CLOSING ACCELERATION (VCDOT/VC)
C =6 FOR NORMAL VELOCITY (VN)
C FOR MULTIPLE FILTERING, THE RIGHTMOST DIGIT INDICATES THE KIND
C OF FILTERING TO BE APPLIED FIRST, THE SECOND RIGHTMOST
C INDICATES THE KIND TO BE APPLIED SECOND, ETC.
C***** FILFAC - ARRAY OF FILTERING FACTORS, IN LEFT-TO-RIGHT ORDER
READ 5515,IPARM,INDFIL,FILFAC
C***** INITIALIZATION
NCATP=NCPM1+1
IQ=1
IR=0
IDIR=1
A7 PG.6 ,9 ,9 ,11,11,12
* * * BEGIN PRODUCTION
10 CONTINUE
* * * READ ONE COMPLETE SCAN OF DATA
CALL
CALL RTAPE(ITAPE)
PAGE 6
B2 PG.12
T
IF(ETAPE) 400, 20
F
20 CONTINUE
* * * CONTINUE READING TAPE UNTIL DESIRE RUN IS FOUND
F
IF(IRUNTP .NE. IRUN)
T
GO TO 10
A7 PG.6
* * * CORRECT RUN
IT = ITTP - 1
* * * CHECK FOR A/C
F
IF(NCRAFT .EQ. 0)
T
GO TO 205
A12 PG.8
* * * PROCESS A/C DATA FOR EVENT ARRAYS
DO 200 I=1,NCRAFT
A11 PG.7
* * * K IS A/C ID

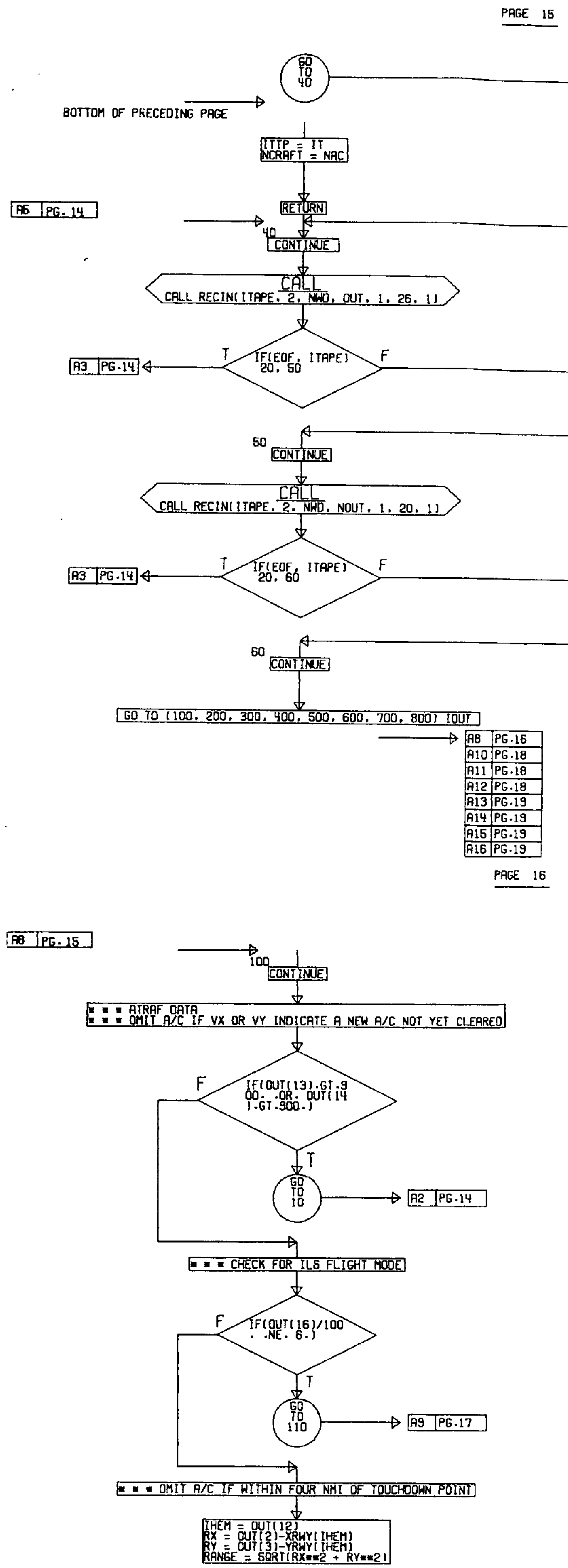
PAGE 7

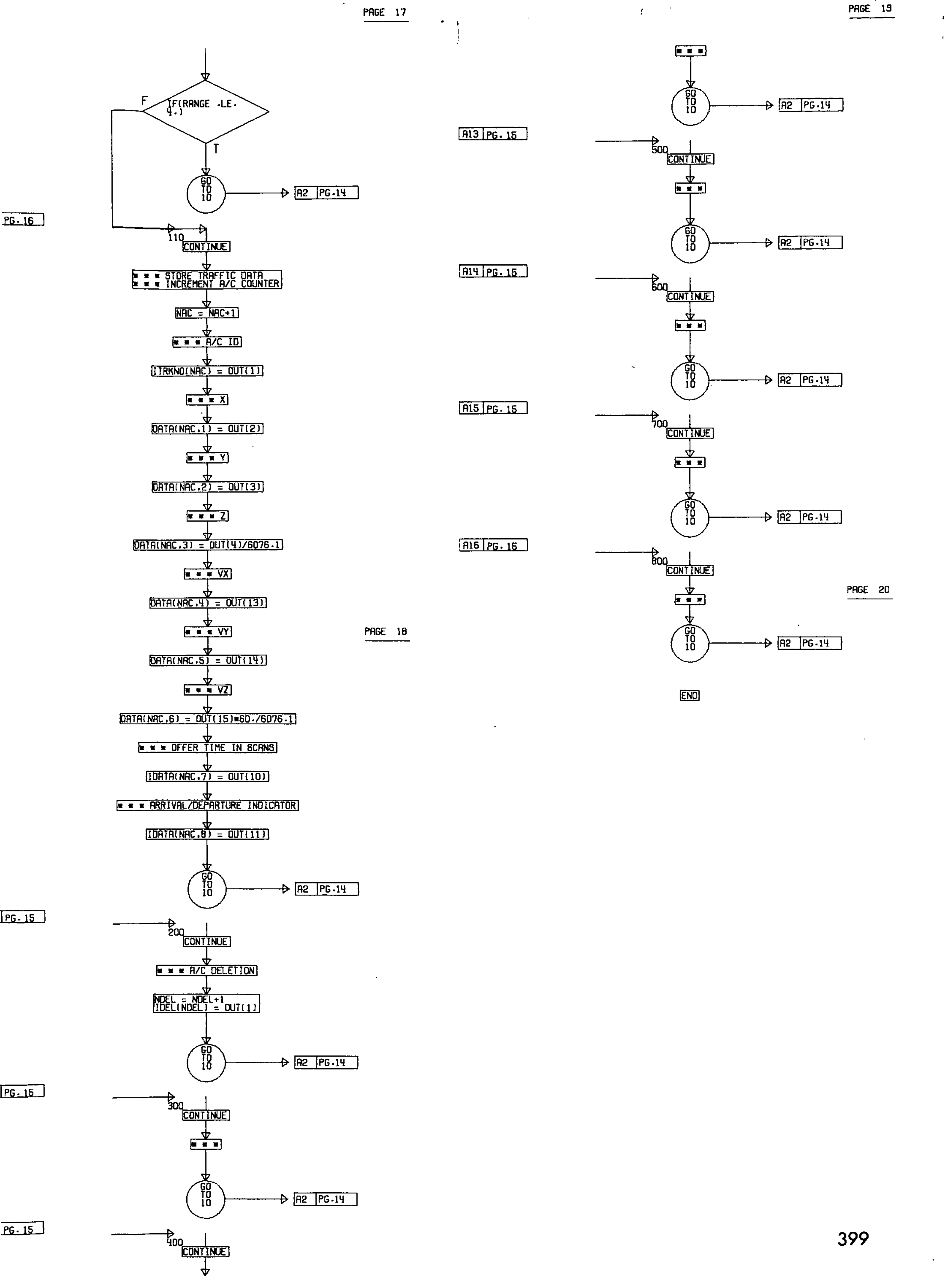

K = ITRKNO(I)
* * * IF THIS A/C HAS APPEARED BEFORE, IEVENT(K) WILL BE GT 0
-
IF(IEVENT(K)) 150, 150, 190
+
0
* * * SET DEPARTURE (1) OR ARRIVAL (2) INDICATOR
150 J = IDATA(I,8)
* * * PLACE INDICATOR IN IEVENT
IEVENT(K) = J
* * * BUMP A/C COUNTER BY TYPE (ARRIVAL OR DEPARTURE)
ITEMP = IKNT(J)+1
IKNT(J) = ITEMP
* * * PLACE A/C ID IN NEVENT
NEVENT(ITEMP, 1, J) = K
* * * PLACE A/C ENTRY SCAN IN NEVENT
NEVENT(ITEMP, 2, J) = IT
* * * PLACE QUEUE ENTRY SCAN (OFFER SCAN) IN NEVENT
NEVENT(ITEMP, 3, J) = IDATA(I,7)
190 CONTINUE
PAGE 8
A11 PG.6
200 CONTINUE
A12 PG.6
205 CONTINUE
* * * CHECK FOR A/C DELETIONS
F
IF(NDEL .LE. 0)
T
GO TO 250
A16 PG.9
DO 230 M=1,NDEL
A15 PG.9
ID = IDEL(M)
DO 210 K=1,2
DO 210 I=1,500
210
F
IF(NEVENT(I,1, K) .EQ. ID)
T
GO TO 220
A14 PG.9
GO TO 230
A15 PG.9
* * * PLACE A/C DELETION SCAN IN NEVENT


PAGE 9
PG.8
PG.8 .8
220
NEVENT(I, 4, K) = IT
230
CONTINUE
PG.8
* * * PROCESS DATA FOR RELATIVE VELOCITIES
* * * MOVE VCS FROM LAST SCAN FROM FIRST TO SECOND
250
CONTINUE
IF(IT .GT. ISTOP)
F
T
GO TO 10
A7 PG.5
IF(IT .LT. ISTART-1)
F
T
GO TO 10
A7 PG.5
DO 260 I=1,50
DO 260 J=1,50
260
VC(I,J,2) = VC(I,J,1)
PAGE 10
* * * COMPUTE VCS FOR THIS SCAN
IF(NCRAFT .EQ. 0)
F
T
GO TO 350
B1 PG.11
DO 300 I=1,NCRAFT
B20 PG.11
DO 300 J=1,NCRAFT
B20 PG.11
* * * RELATIVE POSITION COMPONENTS
DX = DATA(I,1) - DATA(J,1)
DY = DATA(I,2) - DATA(J,2)
DZ = DATA(I,3) - DATA(J,3)
* * * RELATIVE VELOCITY COMPONENTS
DXDT = DATA(I,4) - DATA(J,4)
DYDT = DATA(I,5) - DATA(J,5)
DZDT = DATA(I,6) - DATA(J,6)
* * * RELATIVE RANGE
RIJ = SQRT(DX**2 + DY**2 + DZ**2)
IF(RIJ .NE. 0.)
F
T
GO TO 270
A18 PG.11
VC(I,J,1) = 0.

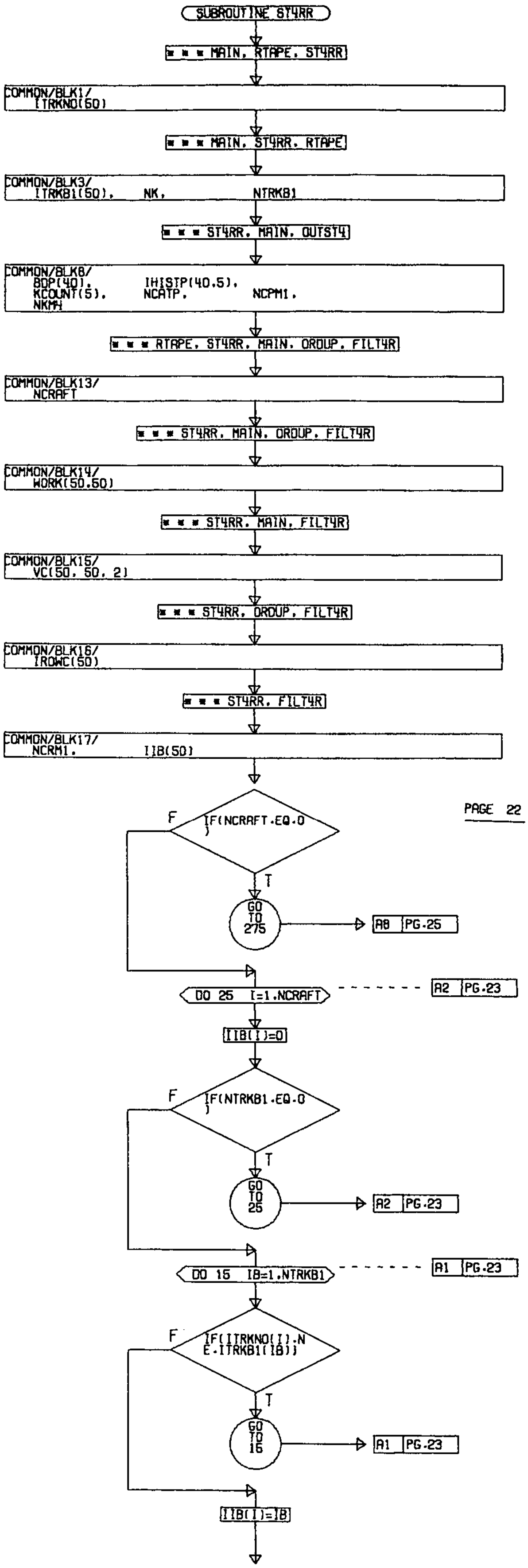

PAGE 11
GO TO 280
A18 PG.10
* * * RELATIVE VELOCITY (KNOTS)
270
VC(I,J,1) = (DX*DXDT + DY*DYDT + DZ*DZDT)/RIJ
* * * IS IT WITHIN THE SCANS OF INTEREST
280
IF(IT .LT. ISTART)
F
T
GO TO 10
A7 PG.5
* * * STORE RELATIVE RANGE AND INCREMENT TOTAL A/C COUNT
RELR(I,J) = RIJ
B20 PG.10 ,10
300
CONTINUE
B1 PG.10
350
IF(IT .LT. ISTART)
F
T
GO TO 10
A7 PG.5
NCTOT = NCTOT+NCRAFT
PAGE 12
* * * ANALYZE RELATIVE RANGES
CALL
CALL ST4RR
GO TO 10
A7 PG.5
B2 PG.6
* * * OUTPUT HISTOGRAM RESULTS
400
CONTINUE
CALL
CALL OUTPT1
CALL
CALL OUTST4
STOP
END

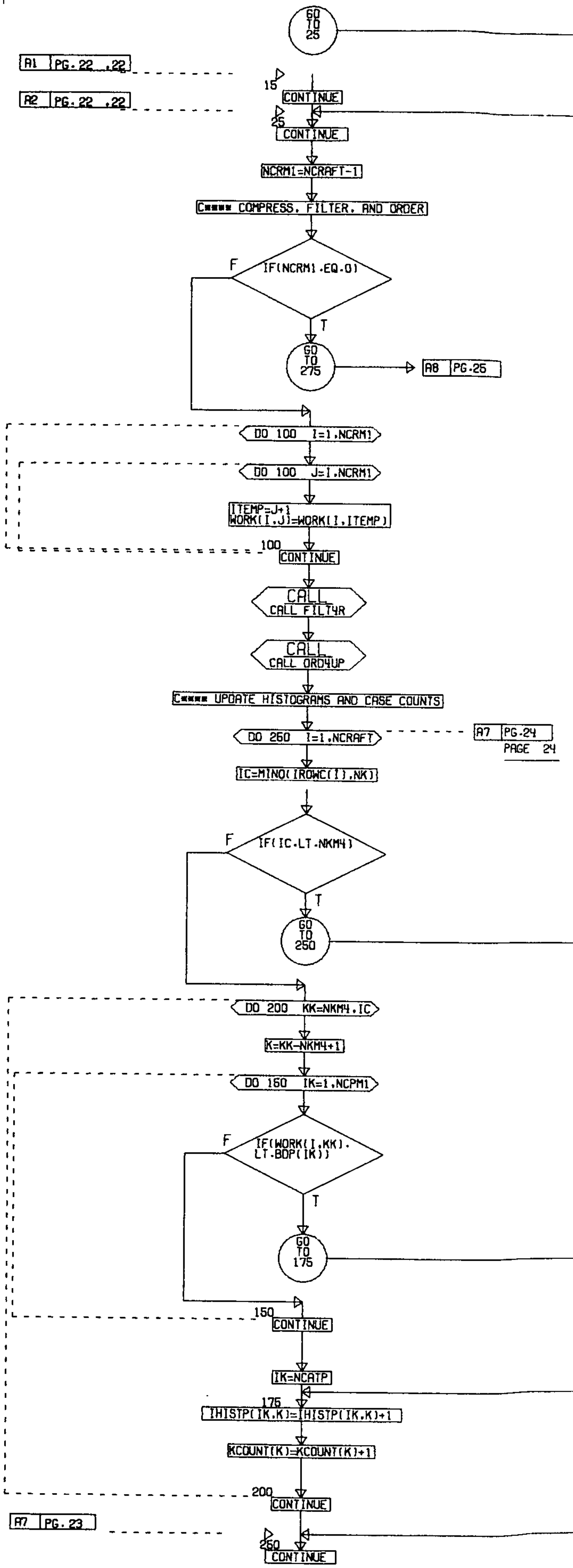

PAGE 13
SUBROUTINE RTAPE(ITAPE)
* * * MAIN, RTAPE, ST4RR
COMMON/BLK1/ ITRKNO(50)
* * * MAIN, RTAPE
COMMON/BLK2/ IDEL(50), IRUNTP, IITP, NDEL, XRWY(2), YRWY(2) , ETAPE
LOGICAL ETAPE
* * * MAIN, ST4RR, RTAPE
COMMON/BLK3/ ITRKB1(50), NK, NTRKB1
* * * MAIN, RTAPE, OUTPT1, ORDER
COMMON/BLK5/ NEVENT(500,4,2)
* * * RTAPE, MAIN, FILT4R
COMMON/BLK9/ DATA(50,8)
* * * RTAPE, ST4RR, MAIN, ORDUP, FILT4R
COMMON/BLK13/ NCRAFT
DIMENSION IDATA(50,8), OUT(26), NOUT(20)
EQUIVALENCE (DATA, IDATA)
NAC = 0
NDEL = 0
C**** SAVE INFO FOR NEXT SCAN
NTRKB1=NCRAFT
PAGE 14
DO 5 J=1,50
5
ITRKB1(J)=ITRKNO(J)
F
IF(ITTP .GT. 1)
T
GO TO 40
A5 PG.15
A2 PG.16 ,17,18,18,18,19
10
CALL
CALL RECIN(ITAPE, 1, NWD, IT, NSCAN, IRUNTP, IOUT)
T
IF(EOF, ITAPE) 20, 30
F
A3 PG.16 ,15
20
CONTINUE
PRINT 25, ITTP
ETAPE = .TRUE.
RETURN
30
CONTINUE
F
IF(IT .EQ. ITTP)
TOP OF NEXT PAGE
T


PAGE 15
GO TO 40
BOTTOM OF PRECEDING PAGE
ITTP = IT
NCRAFT = NAC
RETURN
A6 PG.14
40
CONTINUE
CALL
CALL RECIN(ITAPE, 2, NWD, OUT, 1, 26, 1)
A3 PG.14
T
IF(EOF, ITAPE) 20, 50
F
50
CONTINUE
CALL
CALL RECIN(ITAPE, 2, NWD, NOUT, 1, 20, 1)
A3 PG.14
T
IF(EOF, ITAPE) 20, 60
F
60
CONTINUE
GO TO (100, 200, 300, 400, 500, 600, 700, 800) IOUT
A8 PG.16
A10 PG.18
A11 PG.18
A12 PG.18
A13 PG.19
A14 PG.19
A15 PG.19
A16 PG.19
PAGE 16
A8 PG.15
100
CONTINUE
* * * ATRAF DATA
* * * OMIT A/C IF VX OR VY INDICATE A NEW A/C NOT YET CLEARED
F
IF(OUT(13).GT.900. .OR. OUT(14).GT.900.)
T
GO TO 10
A2 PG.14
* * * CHECK FOR ILS FLIGHT MODE
F
IF(OUT(16)/100. .NE. 6.)
T
GO TO 110
A9 PG.17
* * * OMIT A/C IF WITHIN FOUR NMI OF TOUCHDOWN POINT
IHEM = OUT(12)
RX = OUT(2)-XRWY(IHEM)
RY = OUT(3)-YRWY(IHEM)
RANGE = SQRT(RX**2 + RY**2)



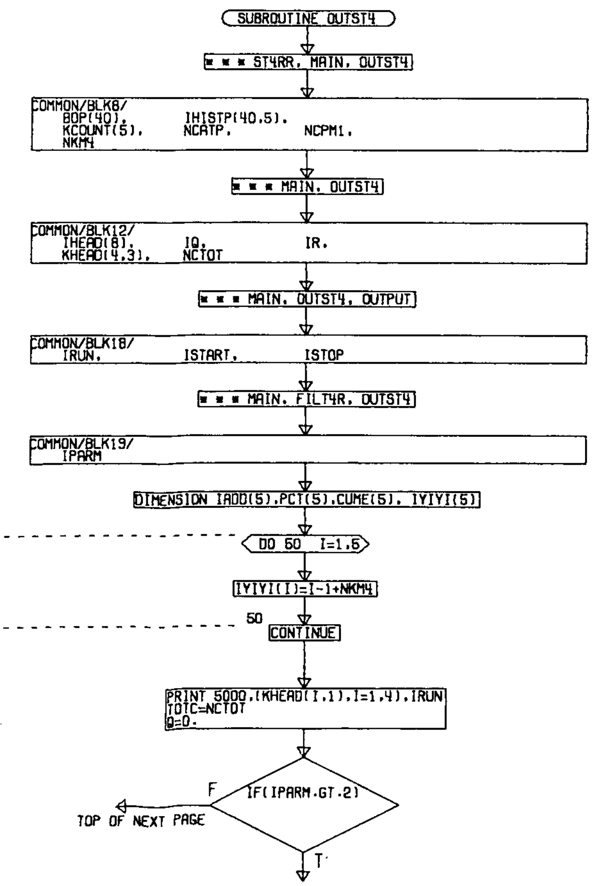
PAGE 17
IF(RANGE .LE. 4.)
F
T
GO TO 10
A2 PG.14
PG.16
110 CONTINUE
* * * STORE TRAFFIC DATA
* * * INCREMENT A/C COUNTER
NAC = NAC+1
* * * A/C ID
ITRKNO(NAC) = OUT(1)
* * * X
DATA(NAC,1) = OUT(2)
* * * Y
DATA(NAC,2) = OUT(3)
* * * Z
DATA(NAC,3) = OUT(4)/6076.1
* * * VX
DATA(NAC,4) = OUT(13)
* * * VY
DATA(NAC,5) = OUT(14)
* * * VZ
DATA(NAC,6) = OUT(15)*60./6076.1
* * * OFFER TIME IN SCANS
IDATA(NAC,7) = OUT(10)
* * * ARRIVAL/DEPARTURE INDICATOR
IDATA(NAC,8) = OUT(11)
GO TO 10
A2 PG.14
PAGE 18
PG.15
200 CONTINUE
* * * A/C DELETION
NDEL = NDEL+1
IDEL(NDEL) = OUT(1)
GO TO 10
A2 PG.14
PG.15
300 CONTINUE
* * *
GO TO 10
A2 PG.14
PG.15
400 CONTINUE
PAGE 19
* * *
GO TO 10
A2 PG.14
A13 PG.15
500 CONTINUE
* * *
GO TO 10
A2 PG.14
A14 PG.15
600 CONTINUE
* * *
GO TO 10
A2 PG.14
A15 PG.15
700 CONTINUE
* * *
GO TO 10
A2 PG.14
A16 PG.15
800 CONTINUE
PAGE 20
* * *
GO TO 10
A2 PG.14
END


SUBROUTINE ST4RR
* * * MAIN, RTAPE, ST4RR
COMMON/BLK1/
ITRKNO(50)
* * * MAIN, ST4RR, RTAPE
COMMON/BLK3/
ITRKB1(50), NK, NTRKB1
* * * ST4RR, MAIN, OUTST4
COMMON/BLK6/
BOP(40), IHISTP(40,5),
KCOUNT(5), NCATP, NCPM1,
NKM4
* * * RTAPE, ST4RR, MAIN, ORDUP, FILT4R
COMMON/BLK13/
NCRAFT
* * * ST4RR, MAIN, ORDUP, FILT4R
COMMON/BLK14/
WORK(50,50)
* * * ST4RR, MAIN, FILT4R
COMMON/BLK15/
VC(50, 50, 2)
* * * ST4RR, ORDUP, FILT4R
COMMON/BLK16/
IROWC(50)
* * * ST4RR, FILT4R
COMMON/BLK17/
NCRM1, IIB(50)
PAGE 22
F
IF(NCRAFT.EQ.0
)
T
GO TO 275
A8 PG.25
DO 25 I=1,NCRAFT
A2 PG.23
IIB(I)=0
F
IF(NTRKB1.EQ.0
)
T
GO TO 25
A2 PG.23
DO 15 IB=1,NTRKB1
A1 PG.23
F
IF(ITRKNO(I).N
E.ITRKB1(IB))
T
GO TO 15
A1 PG.23
IIB(I)=IB


GO TO 25
A1 PG.22 ,22
A2 PG.22 ,22
15
CONTINUE
25
CONTINUE
NCRM1=NCRAFT-1
C**** COMPRESS, FILTER, AND ORDER
F
IF(NCRM1.EQ.0)
T
GO TO 275
A8 PG.25
DO 100 I=1,NCRM1
DO 100 J=I,NCRM1
ITEMP=J+1
WORK(I,J)=WORK(I,ITEMP)
100
CONTINUE
CALL
CALL FILT4R
CALL
CALL ORD4UP
C**** UPDATE HISTOGRAMS AND CASE COUNTS
DO 250 I=1,NCRAFT
A7 PG.24
PAGE 24
IC=MINO(IROWC(I),NK)
F
IF(IC.LT.NKM4)
T
GO TO 250
DO 200 KK=NKM4,IC
K=KK-NKM4+1
DO 150 IK=1,NCPM1
F
IF(WORK(I,KK).
LT.BOP(IK))
T
GO TO 175
150
CONTINUE
IK=NCATP
175
IHISTP(IK,K)=IHISTP(IK,K)+1
KCOUNT(K)=KCOUNT(K)+1
200
CONTINUE
A7 PG.23
260
CONTINUE

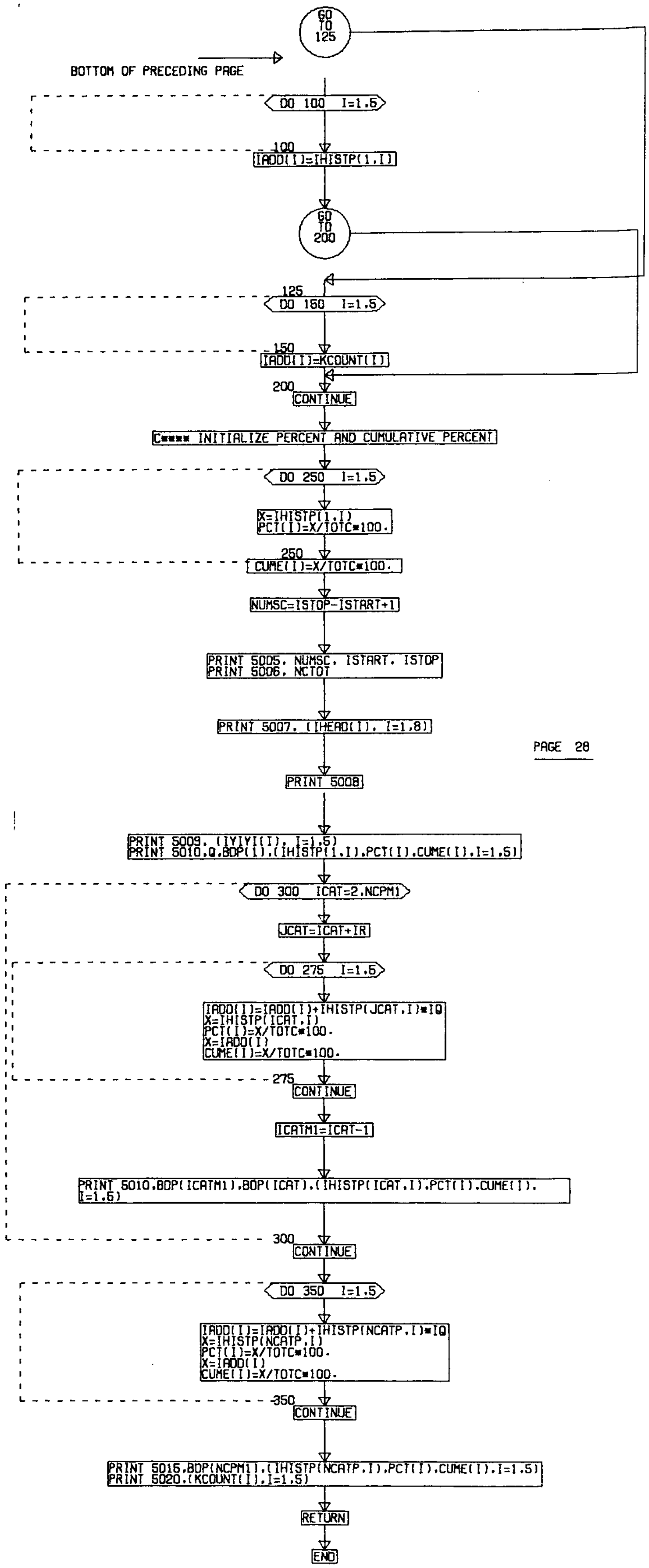
PAGE 25

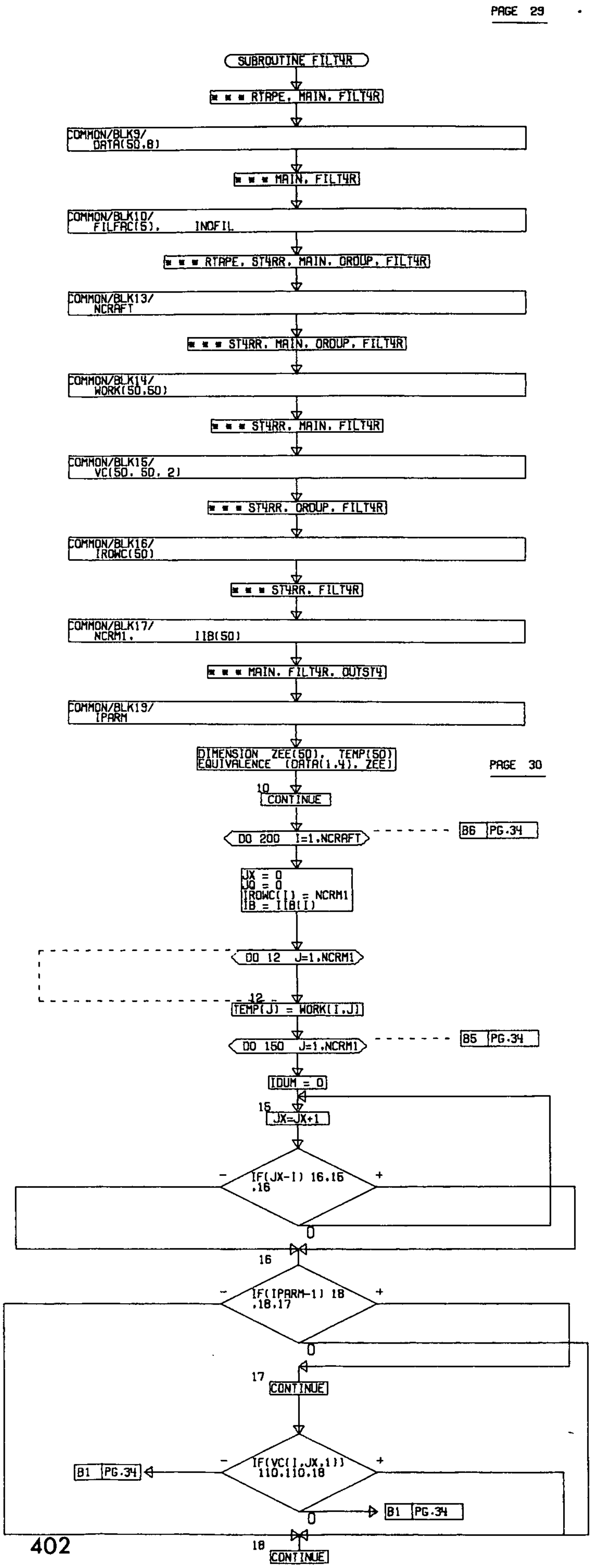
PAGE 26

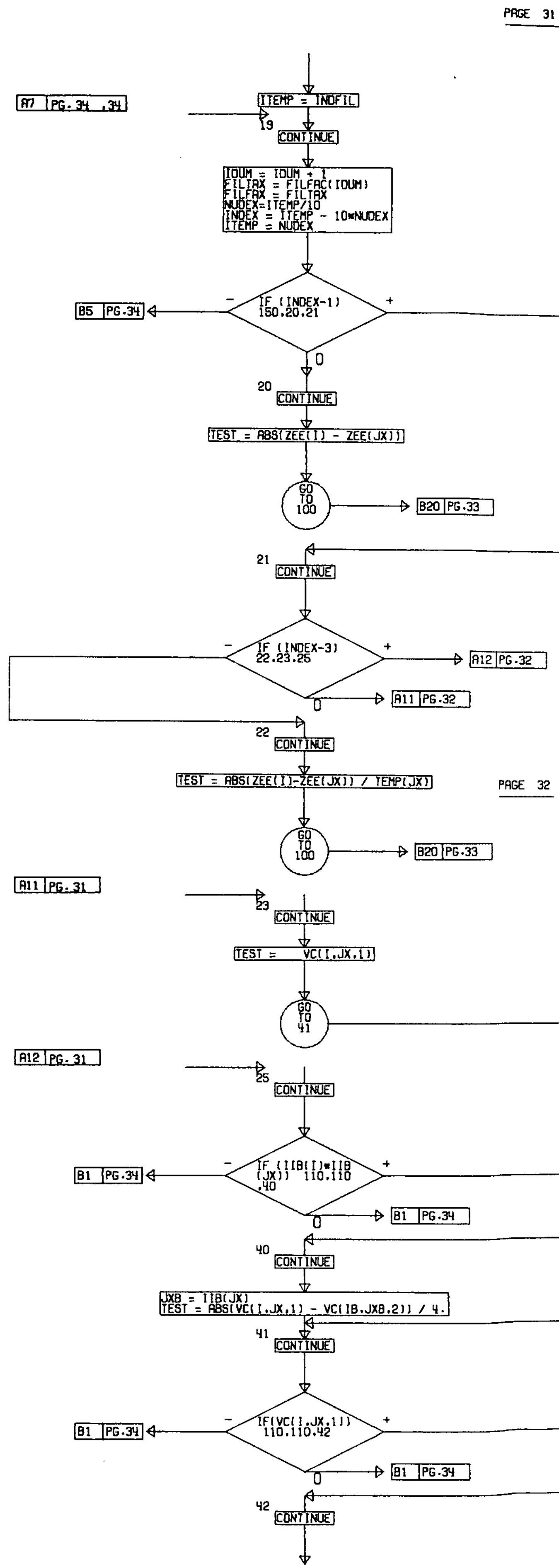


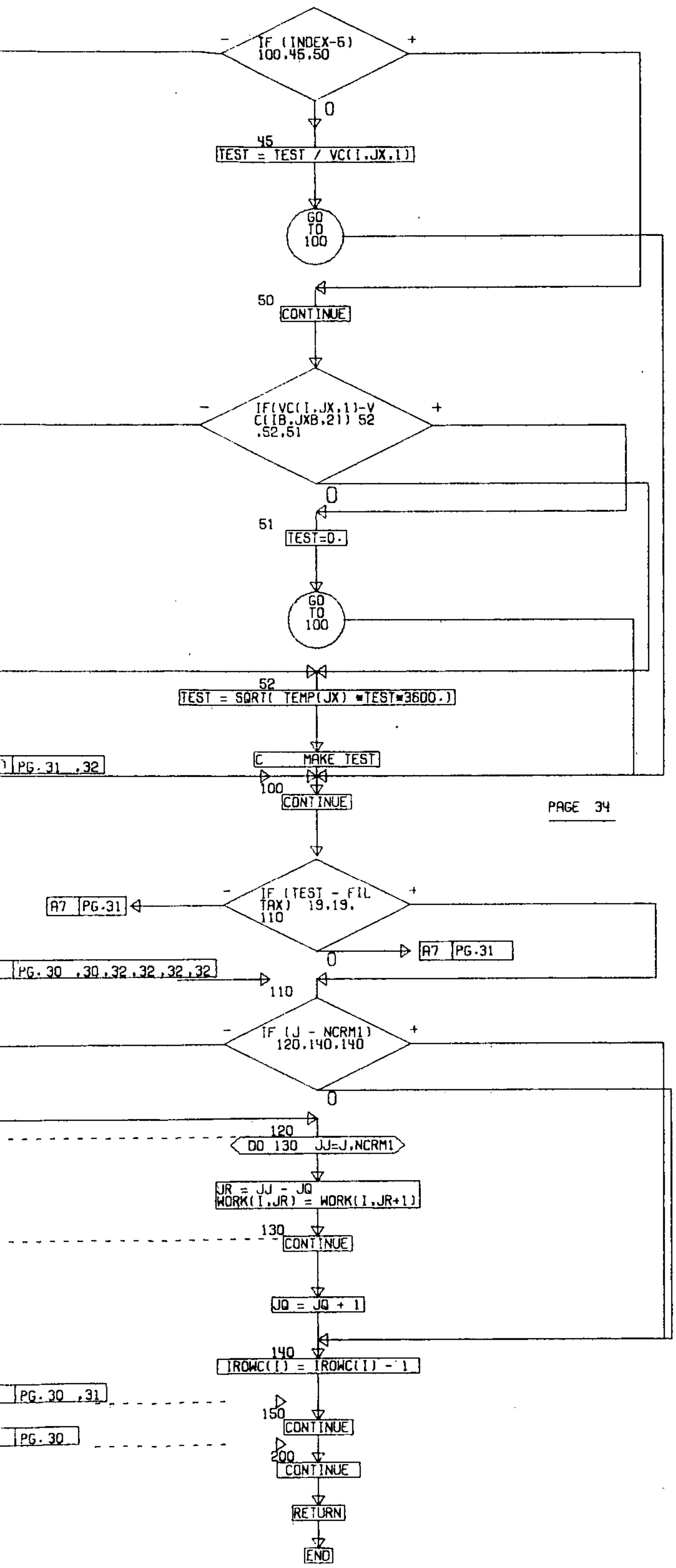
PAGE 27

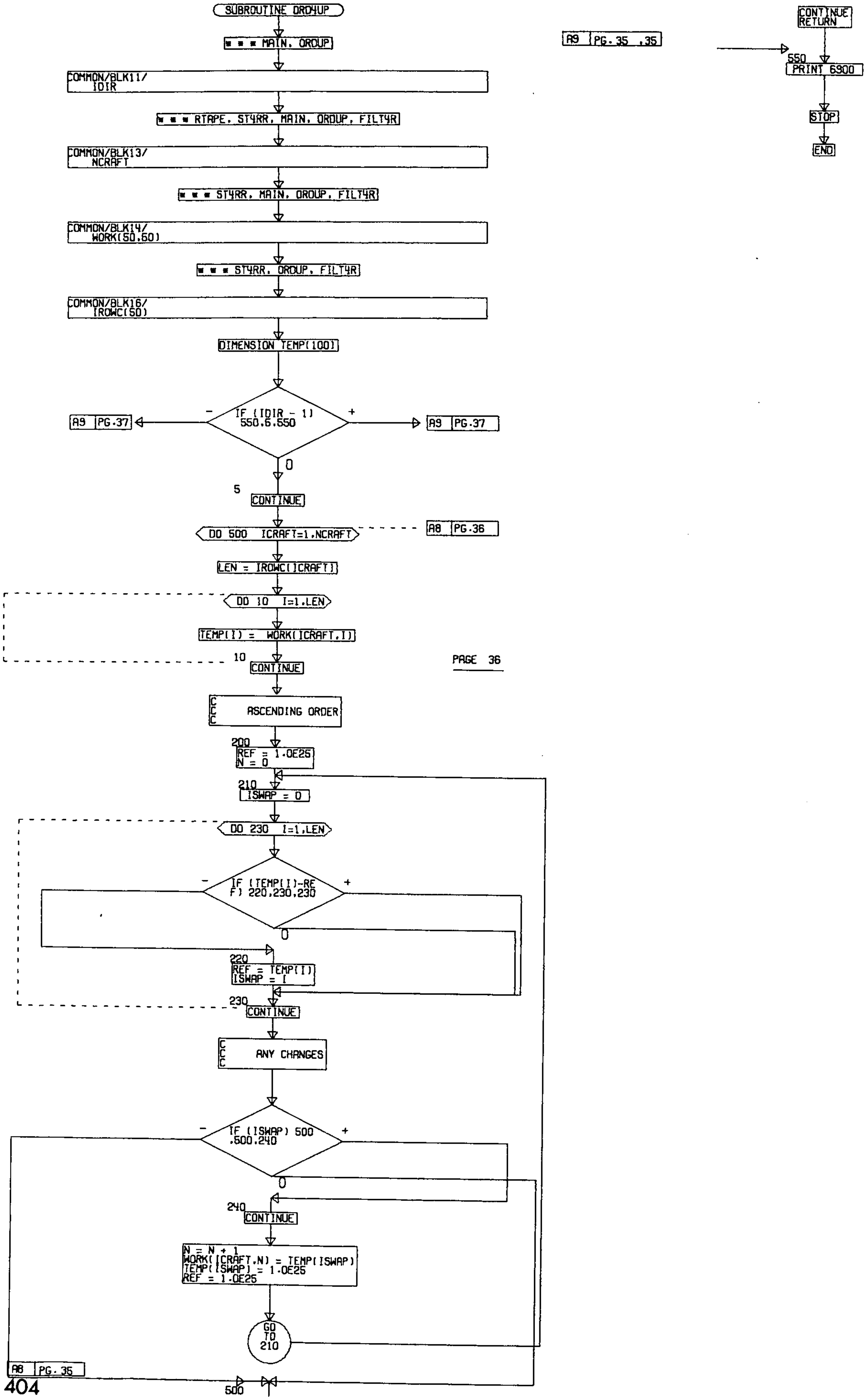
PAGE 28



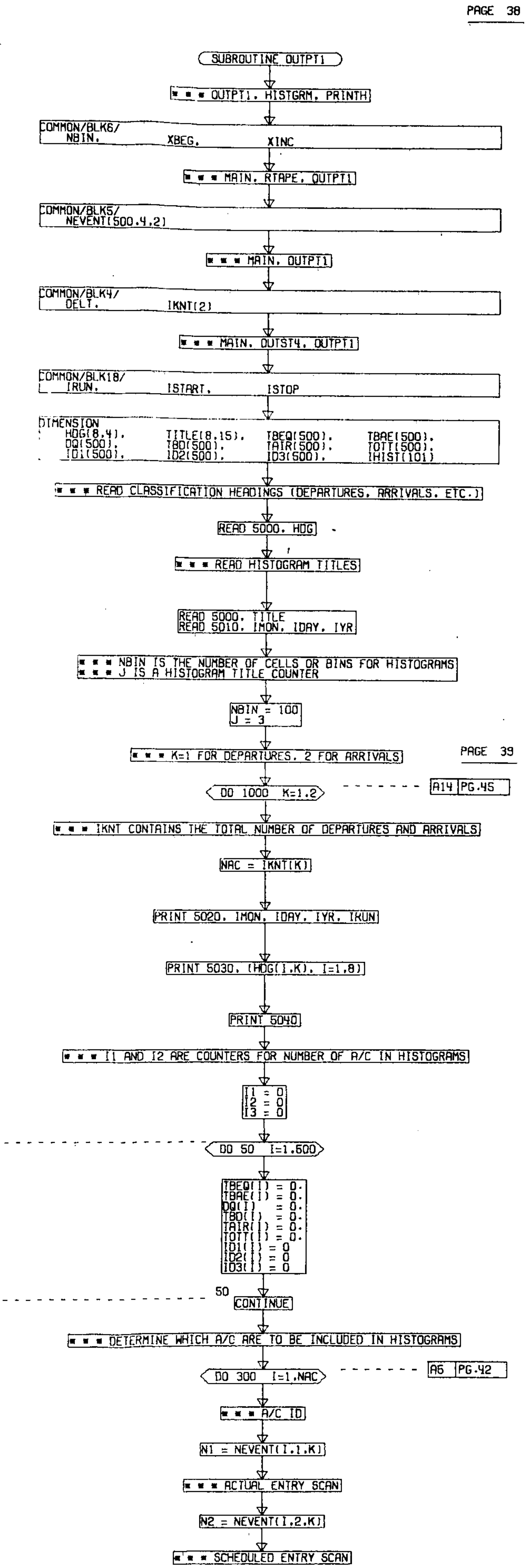

PAGE 29
SUBROUTINE FILT4R
* * * RTAPE, MAIN, FILT4R
COMMON/BLK9/
DATA(50,8)
* * * MAIN, FILT4R
COMMON/BLK10/
FILFAC(5), INDFIL
* * * RTAPE, ST4RR, MAIN, ORDUP, FILT4R
COMMON/BLK13/
NCRAFT
* * * ST4RR, MAIN, ORDUP, FILT4R
COMMON/BLK14/
WORK(50,50)
* * * ST4RR, MAIN, FILT4R
COMMON/BLK15/
VC(50, 50, 2)
* * * ST4RR, ORDUP, FILT4R
COMMON/BLK16/
IROWC(50)
* * * ST4RR, FILT4R
COMMON/BLK17/
NCRM1, IIB(50)
* * * MAIN, FILT4R, OUTST4
COMMON/BLK19/
IPARM
DIMENSION ZEE(50), TEMP(50)
EQUIVALENCE (DATA(1,4), ZEE)
PAGE 30
10
CONTINUE
DO 200 I=1,NCRAFT
B6 PG.34
JX = 0
JQ = 0
IROWC(I) = NCRM1
IB = IIB(I)
DO 12 J=1,NCRM1
12
TEMP(J) = WORK(I,J)
DO 150 J=1,NCRM1
B5 PG.34
IDUM = 0
15
JX=JX+1
IF(JX-I) 16,16,16
16
IF(IPARM-1) 18,18,17
17
CONTINUE
IF(VC(I,JX,1)) 110,110,18
B1 PG.34
B1 PG.34
18
CONTINUE

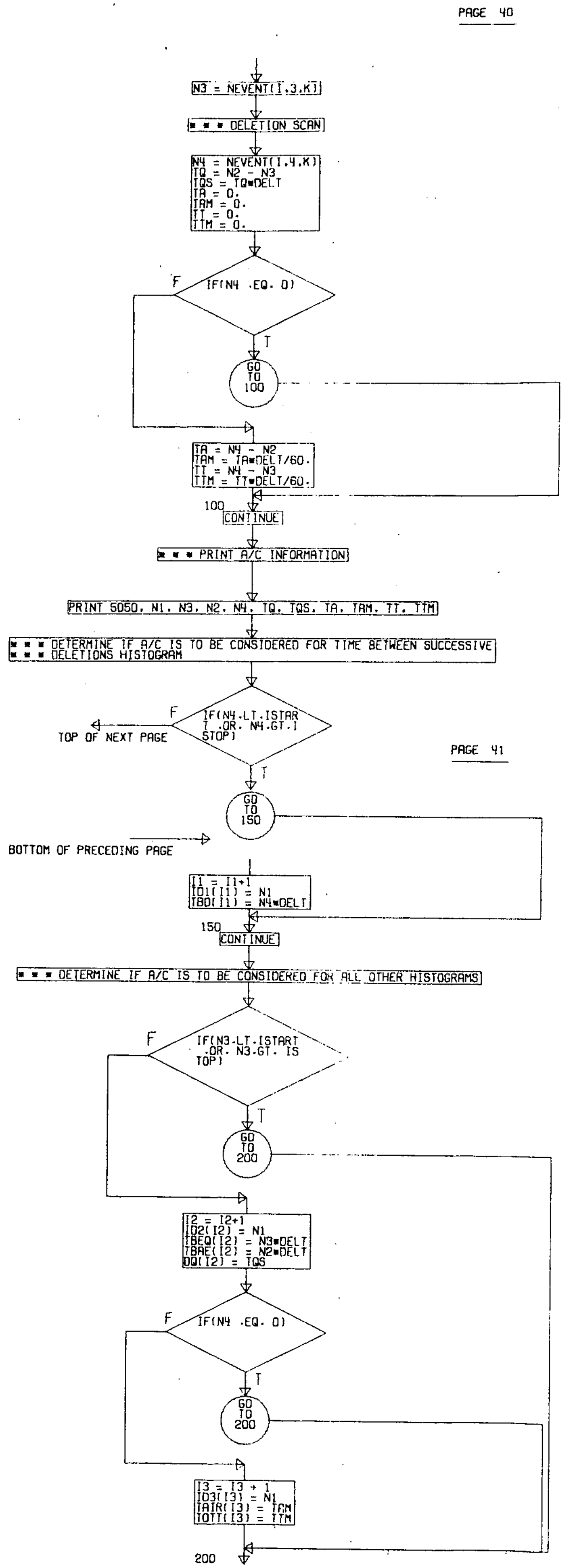

PAGE 31
A7 PG.34 ,34
ITEMP = INDFIL
19
CONTINUE
IDUM = IDUM + 1
FILTAX = FILFAC(IDUM)
FILFAX = FILTAX
NUDEX=ITEMP/10
INDEX = ITEMP - 10*NUDEX
ITEMP = NUDEX
B5 PG.34
IF (INDEX-1) 150,20,21
20
CONTINUE
TEST = ABS(ZEE(I) - ZEE(JX))
GO TO 100
B20 PG.33
21
CONTINUE
IF (INDEX-3) 22,23,25
A12 PG.32
A11 PG.32
22
CONTINUE
TEST = ABS(ZEE(I)-ZEE(JX)) / TEMP(JX)
PAGE 32
GO TO 100
B20 PG.33
A11 PG.31
23
CONTINUE
TEST = VC(I,JX,1)
GO TO 41
A12 PG.31
25
CONTINUE
B1 PG.34
IF (IIB(I)*IIB(JX)) 110,110,40
B1 PG.34
40
CONTINUE
JXB = IIB(JX)
TEST = ABS(VC(I,JX,1) - VC(IB,JXB,2)) / 4.
41
CONTINUE
B1 PG.34
IF(VC(I,JX,1)) 110,110,42
B1 PG.34
42
CONTINUE

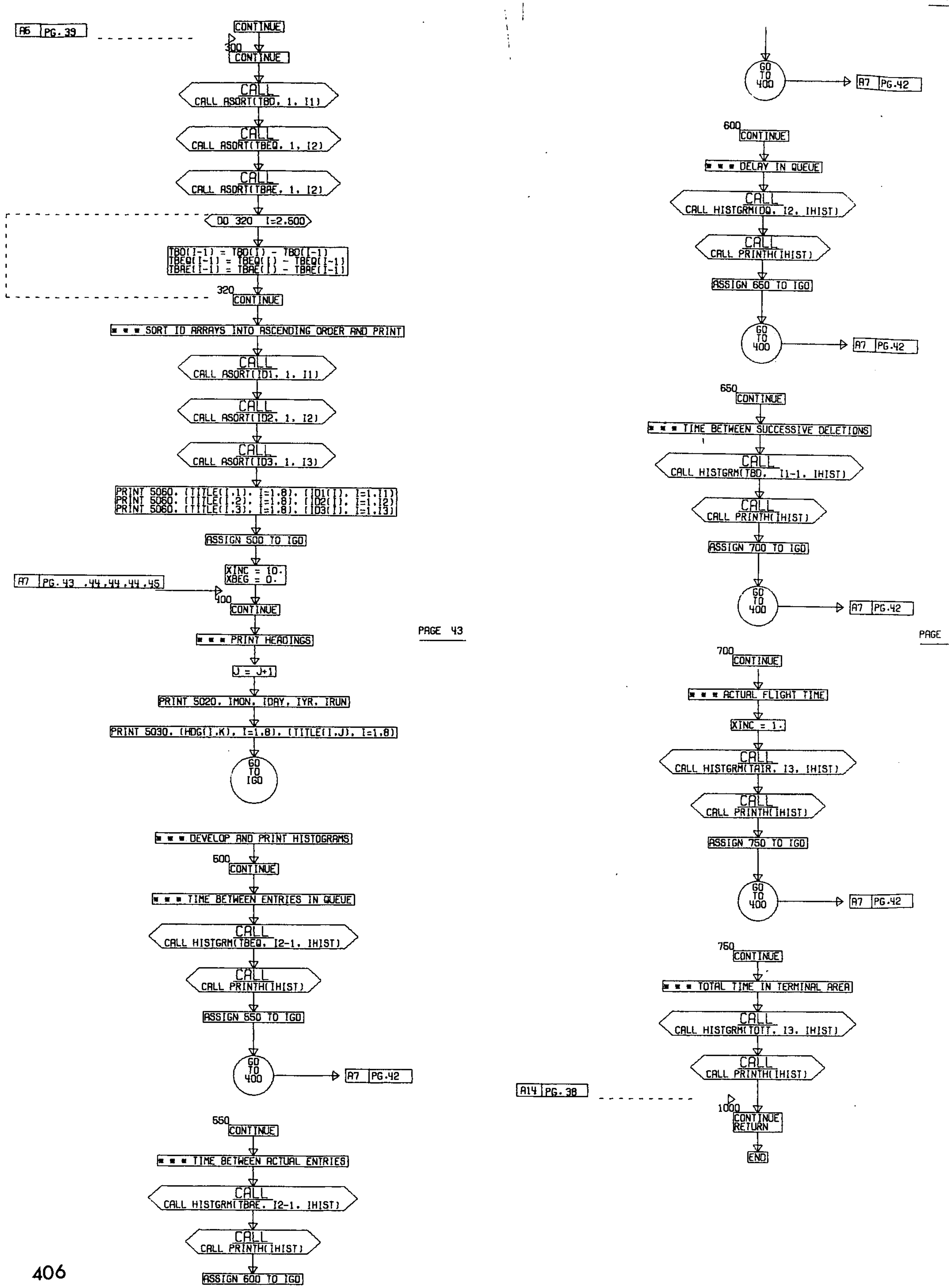

IF (INDEX-5)
100,45,50
45
TEST = TEST / VC(I,JX,1)
GO
TO
100
50
CONTINUE
IF(VC(I,JX,1)-V
C(IB,JXB,2)) 52
,52,51
51
TEST=0.
GO
TO
100
52
TEST = SQRT(TEMP(JX) *TEST*3600.)
C MAKE TEST
PG.31 ,32
100
CONTINUE
PAGE 34
IF (TEST - FIL
TAX) 19,19,
110
A7 PG.31
A7 PG.31
PG.30 ,30,32,32,32,32
110
IF (J - NCRM1)
120,140,140
120
DO 130 JJ=J,NCRM1
JR = JJ - JQ
WORK(I,JR) = WORK(I,JR+1)
130
CONTINUE
JQ = JQ + 1
140
IROWC(I) = IROWC(I) - 1
PG.30 ,31
150
CONTINUE
PG.30
200
CONTINUE
RETURN
END

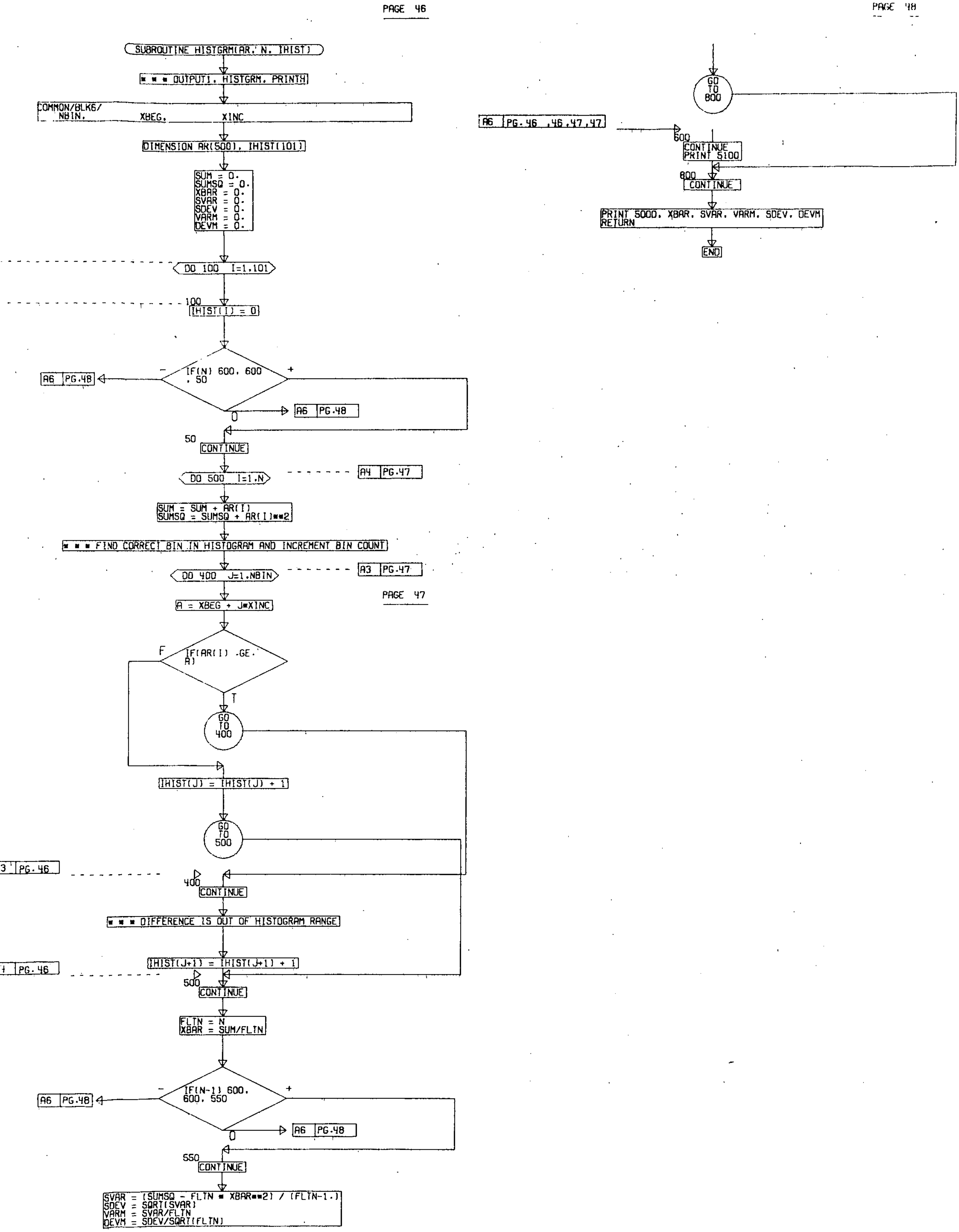


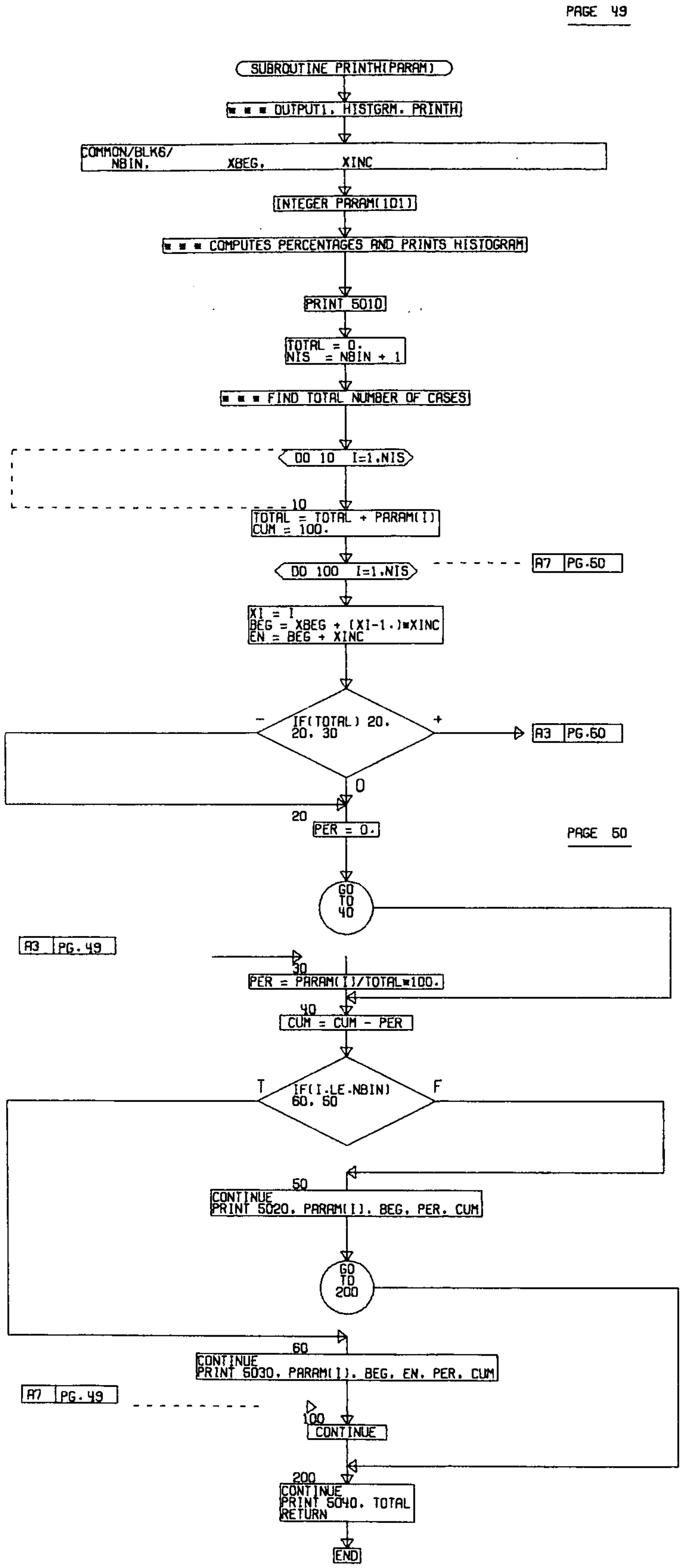
SUBROUTINE ORD4UP
* * * MAIN. ORDUP
COMMON/BLK11/
IDIR
* * * RTAPE. ST4RR. MAIN. ORDUP. FILT4R
COMMON/BLK13/
NCRAFT
* * * ST4RR. MAIN. ORDUP. FILT4R
COMMON/BLK14/
WORK(50.50)
* * * ST4RR. ORDUP. FILT4R
COMMON/BLK16/
IROWC(50)
DIMENSION TEMP(100)
IF (IDIR - 1) 550.6.550
A9 PG.37
A9 PG.37
5
CONTINUE
DO 500 ICRAFT=1.NCRAFT
A8 PG.36
LEN = IROWC(ICRAFT)
DO 10 I=1.LEN
TEMP(I) = WORK(ICRAFT.I)
10
CONTINUE
PAGE 36
C
C ASCENDING ORDER
C
200
REF = 1.0E25
N = 0
210
ISWAP = 0
DO 230 I=1.LEN
IF (TEMP(I)-RE
F) 220.230.230
220
REF = TEMP(I)
ISWAP = I
230
CONTINUE
C
C ANY CHANGES
C
IF (ISWAP) 500
.500.240
240
CONTINUE
N = N + 1
WORK(ICRAFT.N) = TEMP(ISWAP)
TEMP(ISWAP) = 1.0E25
REF = 1.0E25
GO
TO
210
A8 PG. 35
500
A9 PG. 35 .35
CONTINUE
RETURN
550
PRINT 6300
STOP
END


PAGE 38
SUBROUTINE OUTPT1
* * * OUTPT1, HISTGRM, PRINTH
COMMON/BLK6/ NBIN, XBEG, XINC
* * * MAIN, RTAPE, OUTPT1
COMMON/BLK5/ NEVENT(500,4,2)
* * * MAIN, OUTPT1
COMMON/BLK4/ DELT, IKNT(2)
* * * MAIN, OUTST4, OUTPT1
COMMON/BLK18/ IRUN, ISTART, ISTOP
DIMENSION HDG(8,4), TITLE(8,15), TBEQ(500), TBAE(500), DQ(500), TBD(500), TAIR(500), TOTT(500), ID1(500), ID2(500), ID3(500), IHIST(101)
* * * READ CLASSIFICATION HEADINGS (DEPARTURES, ARRIVALS, ETC.)
READ 5000, HDG
* * * READ HISTOGRAM TITLES
READ 5000, TITLE
READ 5010, IMON, IDAY, IYR
* * * NBIN IS THE NUMBER OF CELLS OR BINS FOR HISTOGRAMS
* * * J IS A HISTOGRAM TITLE COUNTER
NBIN = 100
J = 3
PAGE 39
* * * K=1 FOR DEPARTURES, 2 FOR ARRIVALS
DO 1000 K=1,2
A14 PG.45
* * * IKNT CONTAINS THE TOTAL NUMBER OF DEPARTURES AND ARRIVALS
NAC = IKNT(K)
PRINT 5020, IMON, IDAY, IYR, IRUN
PRINT 5030, (HDG(I,K), I=1,8)
PRINT 5040
* * * I1 AND I2 ARE COUNTERS FOR NUMBER OF A/C IN HISTOGRAMS
I1 = 0
I2 = 0
I3 = 0
DO 50 I=1,600
TBEQ(I) = 0.
TBAE(I) = 0.
DQ(I) = 0.
TBD(I) = 0.
TAIR(I) = 0.
TOTT(I) = 0.
ID1(I) = 0
ID2(I) = 0
ID3(I) = 0
50
CONTINUE
* * * DETERMINE WHICH A/C ARE TO BE INCLUDED IN HISTOGRAMS
DO 300 I=1,NAC
A6 PG.42
* * * A/C ID
N1 = NEVENT(I,1,K)
* * * ACTUAL ENTRY SCAN
N2 = NEVENT(I,2,K)
* * * SCHEDULED ENTRY SCAN


PAGE 40
N3 = NEVENT(I,3,K)
* * * DELETION SCAN
N4 = NEVENT(I,4,K)
TQ = N2 - N3
TQS = TQ*DELT
TA = 0.
TAM = 0.
TT = 0.
TTM = 0.
IF(N4 .EQ. 0)
F
T
GO TO 100
TA = N4 - N2
TAM = TA*DELT/60.
TT = N4 - N3
TTM = TT*DELT/60.
100
CONTINUE
* * * PRINT A/C INFORMATION
PRINT 5050, N1, N3, N2, N4, TQ, TQS, TA, TAM, TT, TTM
* * * DETERMINE IF A/C IS TO BE CONSIDERED FOR TIME BETWEEN SUCCESSIVE
* * * DELETIONS HISTOGRAM
IF(N4.LT.ISTART .OR. N4.GT.ISTOP)
F
TOP OF NEXT PAGE
PAGE 41
T
GO TO 150
BOTTOM OF PRECEDING PAGE
I1 = I1+1
ID1(I1) = N1
TBD(I1) = N4*DELT
150
CONTINUE
* * * DETERMINE IF A/C IS TO BE CONSIDERED FOR ALL OTHER HISTOGRAMS
IF(N3.LT.ISTART .OR. N3.GT. ISTOP)
F
T
GO TO 200
I2 = I2+1
ID2(I2) = N1
TBEQ(I2) = N3*DELT
TBAE(I2) = N2*DELT
DQ(I2) = TQS
IF(N4 .EQ. 0)
F
T
GO TO 200
I3 = I3 + 1
ID3(I3) = N1
TAIR(I3) = TAM
TOTT(I3) = TTM
200


A6 PG. 39
CONTINUE
300
CONTINUE
CALL
CALL ASORT(TBD, 1, I1)
CALL
CALL ASORT(TBEQ, 1, I2)
CALL
CALL ASORT(TBAE, 1, I2)
DO 320 I=2,500
TBD(I-1) = TBD(I) - TBD(I-1)
TBEQ(I-1) = TBEQ(I) - TBEQ(I-1)
TBAE(I-1) = TBAE(I) - TBAE(I-1)
320
CONTINUE
* * * SORT ID ARRAYS INTO ASCENDING ORDER AND PRINT
CALL
CALL ASORT(ID1, 1, I1)
CALL
CALL ASORT(ID2, 1, I2)
CALL
CALL ASORT(ID3, 1, I3)
PRINT 5060, (TITLE(I,1), I=1,8), (ID1(I), I=1,I1)
PRINT 5060, (TITLE(I,2), I=1,8), (ID2(I), I=1,I2)
PRINT 5060, (TITLE(I,3), I=1,8), (ID3(I), I=1,I3)
ASSIGN 500 TO IGO
XINC = 10.
XBEG = 0.
A7 PG. 43 ,44,44,44,45
400
CONTINUE
PAGE 43
* * * PRINT HEADINGS
J = J+1
PRINT 5020, IMON, IDAY, IYR, IRUN
PRINT 5030, (HDG(I,K), I=1,8), (TITLE(I,J), I=1,8)
GO TO IGO
* * * DEVELOP AND PRINT HISTOGRAMS
500
CONTINUE
* * * TIME BETWEEN ENTRIES IN QUEUE
CALL
CALL HISTGRM(TBEQ, I2-1, IHIST)
CALL
CALL PRINTH(IHIST)
ASSIGN 550 TO IGO
GO TO 400
A7 PG.42
550
CONTINUE
* * * TIME BETWEEN ACTUAL ENTRIES
CALL
CALL HISTGRM(TBAE, I2-1, IHIST)
CALL
CALL PRINTH(IHIST)
ASSIGN 600 TO IGO
PAGE
GO TO 400
A7 PG.42
600
CONTINUE
* * * DELAY IN QUEUE
CALL
CALL HISTGRM(DQ, I2, IHIST)
CALL
CALL PRINTH(IHIST)
ASSIGN 650 TO IGO
GO TO 400
A7 PG.42
650
CONTINUE
* * * TIME BETWEEN SUCCESSIVE DELETIONS
CALL
CALL HISTGRM(TBD, I1-1, IHIST)
CALL
CALL PRINTH(IHIST)
ASSIGN 700 TO IGO
GO TO 400
A7 PG.42
PAGE
700
CONTINUE
* * * ACTUAL FLIGHT TIME
XINC = 1.
CALL
CALL HISTGRM(TAIR, I3, IHIST)
CALL
CALL PRINTH(IHIST)
ASSIGN 750 TO IGO
GO TO 400
A7 PG.42
750
CONTINUE
* * * TOTAL TIME IN TERMINAL AREA
CALL
CALL HISTGRM(TOTT, I3, IHIST)
CALL
CALL PRINTH(IHIST)
A14 PG. 38
1000
CONTINUE
RETURN
END


PAGE 46
SUBROUTINE HISTGRM(AR, N, IHIST)
* * * OUTPUT1, HISTGRM, PRINTH
COMMON/BLK6/ NBIN, XBEG, XINC
DIMENSION AR(500), IHIST(101)
SUM = 0.
SUMSQ = 0.
XBAR = 0.
SVAR = 0.
SDEV = 0.
VARM = 0.
DEVM = 0.
DO 100 I=1,101
100
IHIST(I) = 0
IF(N) 600, 600, 50
A6 PG.48
A6 PG.48
50
CONTINUE
DO 500 I=1,N
A4 PG.47
SUM = SUM + AR(I)
SUMSQ = SUMSQ + AR(I)**2
* * * FIND CORRECT BIN IN HISTOGRAM AND INCREMENT BIN COUNT
DO 400 J=1,NBIN
A3 PG.47
PAGE 47
A = XBEG + J*XINC
IF(AR(I) .GE. A)
F
T
GO TO 400
IHIST(J) = IHIST(J) + 1
GO TO 500
3 PG.46
400
CONTINUE
* * * DIFFERENCE IS OUT OF HISTOGRAM RANGE
IHIST(J+1) = IHIST(J+1) + 1
PG.46
500
CONTINUE
FLTN = N
XBAR = SUM/FLTN
IF(N-1) 600, 600, 550
A6 PG.48
A6 PG.48
550
CONTINUE
SVAR = (SUMSQ - FLTN * XBAR**2) / (FLTN-1.)
SDEV = SQRT(SVAR)
VARM = SVAR/FLTN
DEVM = SDEV/SQRT(FLTN)
PAGE 48
GO TO 800
A6 PG.46 ,46,47,47
600
CONTINUE
PRINT 5100
800
CONTINUE
PRINT 5000, XBAR, SVAR, VARM, SDEV, DEVM
RETURN
END


PAGE 49
SUBROUTINE PRINTH(PARAM)
* * * OUTPUT1, HISTGRM, PRINTH
COMMON/BLK6/ NBIN, XBEG, XINC
INTEGER PARAM(101)
* * * COMPUTES PERCENTAGES AND PRINTS HISTOGRAM
PRINT 5010
TOTAL = 0.
NIS = NBIN + 1
* * * FIND TOTAL NUMBER OF CASES
DO 10 I=1,NIS
10
TOTAL = TOTAL + PARAM(I)
CUM = 100.
DO 100 I=1,NIS
A7 PG.50
XI = I
BEG = XBEG + (XI-1.)*XINC
EN = BEG + XINC
IF(TOTAL) 20, 20, 30
-
+
A3 PG.50
0
20
PER = 0.
PAGE 50
GO TO 40
A3 PG.49
30
PER = PARAM(I)/TOTAL*100.
40
CUM = CUM - PER
IF(I.LE.NBIN) 60, 50
T
F
50
CONTINUE
PRINT 5020, PARAM(I), BEG, PER, CUM
GO TO 200
60
CONTINUE
PRINT 5030, PARAM(I), BEG, EN, PER, CUM
A7 PG.49
100
CONTINUE
200
CONTINUE
PRINT 5040, TOTAL
RETURN
END

VIII. REFERENCES

[1] Britt, C. L. Definition of a Terminal Area Air Traffic Model for Studies of Advanced Instrumentation and Control Techniques. NASA CR-111979, Research Triangle Institute, Research Triangle Park, North Carolina, December 1971.

[2] Britt, C. L., T. M. Walsh, J. A. Modi and E. G. Baxa. An Investigation of Vehicle Dependent Aspects of Terminal Area ATC Operation. Presented at the 1972 Joint Automatic Control Conference, Palo Alto, California, August 16-18, 1972.

[3] Conway, Johnson and Maxwell. Some Problems of Digital Systems Simulation. Management Science, Vol. 5, 1959, page 92.

[4] Britt, C. L. Statistical Evaluation of Aircraft Collision-Hazard Warning System Techniques in the Terminal Area--Phase II. NASA CR-1470, Research Triangle Institute, Research Triangle Park, North Carolina, December 1969.